Contents

NoteBook ... **3**

⊠⊠⊠⊠⊠⊠⊠ ... **4**

- ⊠⊠⊠⊠ ... 5
- NoC ⊠⊠⊠⊠ ... 14
- ⊠⊠⊠⊠⊠ ... 22
- SoC ... 31
- ⊠⊠ ... 33
- ⊠⊠ ... 54
- ⊠⊠⊠⊠⊠⊠⊠ ... 60
- Memory consistency and cache coherence ... 64

⊠⊠ ... **69**

- RISC-V ⊠⊠⊠⊠⊠⊠ ... 72
- Cache as RAM & On chip memory ... 73
- EDK2 System Table ... 74
- EDK2 Event ... 76
- EDK2 Memory Map ... 89

OS ... **90**

⊠⊠⊠⊠⊠⊠⊠ ... **92**

- ACPI ... 93
- CEDT ... 102

AMBA ... **105**

PCIe ... **106**

- ARM N2 ⊠⊠⊠⊠ ... 121
- PCIe ⊠⊠⊠⊠⊠⊠⊠ ... 123
- PCIe ⊠⊠⊠⊠ ... 129

CXL ... **140**

- CXL ⊠⊠⊠ ... 143
- CXL _in _CMN ... 145
- CXL _in _SCP ... 147
- CXL ⊠⊠ ... 152

DDR ... **153**

⊠⊠⊠⊠⊠⊠⊠ ... **158**

- LeetCode ... 159
- C ⊠⊠⊠⊠ ... 159
- Bit Map ... 163
- ⊠⊠⊠ ... 164
- ⊠⊠ ... 168
- ⊠⊠⊠ ... 178

- ☒☒☒ 180
- ☒☒☒ 183
- ☒ 185

System Design 189

- System Design Overview 189

Tool 190

- Git 190

☒☒ 191

- ☒☒☒☒☒☒ 192
- 2024 Reading 193

NoteBook

- [illegible]
- [illegible]
- [illegible]
- [illegible]

by Zhenqiu

☒☒☒☒☒☒☒

[illegible]

[illegible]

[illegible]

[toc]

[illegible]

[illegible] [illegible] [illegible] [illegible]

image-20240711133545527

[illegible] [illegible] [illegible]

[illegible]

- [illegible]A [illegible] B [illegible]
- [illegible]
- [illegible] [illegible]

[illegible]

[illegible]

image-20240711134934686

[illegible]

[illegible]

[illegible]

[illegible] [illegible]

[illegible]

image-20240711140707047

4G [illegible]4K [illegible] 2^20 [illegible] 4B[illegible] 4M [illegible]

[illegible] 4M [illegible] table[illegible] table [illegible]

[illegible]

[illegible] 32 [illegible] 4 GB[illegible] 512 MB [illegible]

image-20240711141407243

[illegible] 14.5 [illegible] VA[31:20][illegible] 12 [illegible] 4096 [illegible] VA[19:12][illegible] 8 [illegible] 256 [illegible] **16 KB** [illegible] 4096 [illegible] 4 [illegible] 16 KB[illegible] PFN [illegible]

[illegible] entry [illegible]5 [illegible] KB[illegible]

[illegible]

ARM64 [illegible] 3 [illegible] 4 [illegible]

X86 [illegible] 5 [illegible]

ARM64 [illegible]

ARM64 [illegible] MMU [illegible] TLB [illegible] Table Walk Unit[illegible] TWU[illegible]

image-20240711143546582

[illegible] SMP[illegible] Symmetric Multi-Processor[illegible] MMU [illegible] TLB [illegible]

image-20240711144032889

[illegible]

AArch64 [illegible] MMU [illegible]

[illegible] VA[illegible] PA[illegible]

[illegible] 1[illegible] Intermediate Physical Address[illegible] IPA[illegible] 2[illegible] IPA [illegible] PA[illegible]

ARMv8 [illegible] 4 KB[illegible] 16 KB [illegible] 64 KB [illegible] 3 [illegible]

[illegible] AArch64 [illegible] 48 [illegible] VA [illegible] 256 TB[illegible]

- [illegible] 0x0000000000000000 [illegible] 0x0000FFFFFFFFFFFF[illegible]

[illegible] 0[illegible] **TTBR0 _ELx** [illegible]

- [illegible] 0xFFFF000000000000 [illegible] 0xFFFFFFFFFFFFFFFF[illegible] 1[illegible] **TTBR1 _ELx** [illegible]

image-20240711145449359

48 [illegible]4K [illegible]

image-20240711145404168

[illegible]

[illegible] x86 _64 [illegible] AArch64 [illegible] 14.16 [illegible] 3 [illegible] CPU [illegible] MMU [illegible] TTBR0 [illegible] CPU [illegible] MMU [illegible] TTBR1 [illegible]

image-20240711145908203

[illegible]

AArch64 [illegible] L0[illegible]L3 [illegible] L0[illegible]L2 [illegible] 14.17 [illegible]

image-20240711150047513

L3 Entry

image-20240711150252333

[illegible]

[illegible] ù [illegible] Translation Control Register[illegible] TCR[illegible] ù [illegible] System Control Register[illegible] SCTLR[illegible] ù [illegible] Translation Table Base Register[illegible] TTBR[illegible]

[illegible]

[illegible] VA[illegible] = [illegible] PA[illegible]

image-20240711150627716

SMMU

[toc]

[illegible] SMMU[illegible]

[illegible] (SMMU, System Memory Management Unit) [illegible]SMMU [illegible]

1. [illegible]- [illegible]SMMU [illegible]

- **I/O** [illegible]SMMU [illegible] I/O [illegible]

1. [illegible]- [illegible]SMMU [illegible]

- [illegible]SMMU [illegible]

1. [illegible]- [illegible]SMMU [illegible] I/O [illegible]
2. [illegible]- [illegible]SMMU [illegible]

- [illegible]SMMU [illegible]

[illegible]

MMU-700 [illegible]

[illegible] ARM [illegible]SMMU [illegible] IO [illegible] IOMMU [illegible] MMU-700 [illegible] IP [illegible]

[illegible] SMMU Spec[illegible] MMU-700 TRM[illegible]

MMU [illegible] SMMU [illegible]

image-20240704172717749

SMMU [illegible]

image-20240704172835264

1[illegible] SMMU [illegible] DMA[illegible]device [illegible]device1 [illegible] VA [illegible]smmu [illegible] VA [illegible] PA[illegible]

2[illegible] SMMU [illegible] PCIe Root Complex (which itself hosts a network of endpoints)[illegible] PCIE [illegible] ATS[illegible]

image-20240704172210004

1[illegible]SMMU A [illegible] complex device [illegible]

2[illegible]SMMU B [illegible] device[illegible] I/O[illegible] DMA[illegible]

3[illegible]SMMU C [illegible]

a. central translation table walker[illegible]

b. [illegible] device [illegible] PCIe [illegible]

image-20240704173417659

MMU-700 [illegible]

image-20240704174121552

image-20240704174321677

The MMU-700 contains the following key components: **Translation Buffer Unit (TBU)** The TBU contains Translation Lookaside Buffers (TLBs) that cache translation tables. The MMU-700 implements a TBU that can be connected to single master or multiple masters. It is also possible to connect multiple TBUs to a single master to improve performance. These TBUs are local to the corresponding master and can be one of the following:

- ACE-Lite TBU
- LTI TBU

Translation Control Unit (TCU) The TCU controls and manages the address translations. The MMU-700 implements a single TCU. In MMU-700-based systems, the AMBA® DTI protocol defines the standard for communicating with the TCU. See the AMBA® DTI Protocol Specification.

DTI interconnect The DTI interconnect connects multiple TBUs to the TCU

TBU [illegible]**TBU ACE [illegible] PA [illegible] transaction [illegible] c2000 [illegible] TBU LTI [illegible] transaction [illegible] PA [illegible] pcie controller**

image-20240704184219326

image-20240704183355432

image-20240704183407823

TLB [illegible]

The TBU contains Translation Lookaside Buffers (TLBs) that cache translation tables.

TBU [illegible] cache[illegible] micro TLB [illegible] mainTLB[illegible]

[illegible]

PA = F(streamID, substreamID, Address)

step1.master[illegible] pcie controller[illegible] VA [illegible] transaction

step2.SMMU [illegible] VA[illegible] SMMU _CRO.SMMUEN [illegible] SMMU,[illegible] bypass,[illegible] VA [illegible] PA [illegible] DMA [illegible],[illegible] StreamlD [illegible] STE;

[illegible] STE.Config [illegible] Stage1 [illegible],[illegible] VA [illegible] IPA [illegible] Stage2 [illegible],[illegible] STE.S1ContextPtr [illegible] SubStreamID [illegible] CD,[illegible] Stage 1 translation table [illegible] ASID,[illegible] Stage 1 translation table walk,[illegible] VA [illegible] IPA [illegible] Stage 2 [illegible];

[illegible] STE.Config [illegible] Stage2 [illegible],[illegible] IPA [illegible] PA [illegible] SMMU [illegible],[illegible] STE.S2TTB [illegible] Stage 2 translation table [illegible] VMID,[illegible] Stage 2 translation table walk,[illegible] IPA [illegible] PA [illegible] SMMU [illegible],[illegible]

Note:[illegible] Stage 1/2 translation table walk [illegible],[illegible] SMMU [illegible] TLB [illegible],[illegible],[illegible] translation table walk[illegible] translation table walk [illegible] TLB [illegible]

[illegible] untranslated transaction, PCIe [illegible] ATS translation request/PRI [illegible]

[illegible] PA

- TBU [illegible] TLB [illegible] PA
- TBU [illegible] TLB [illegible]TBU [illegible] TCU [illegible] transaction table walk[illegible] memory [illegible] translation table [illegible] VA [illegible] PA [illegible]

step3.SMMU [illegible] PA [illegible] transaction

image-20240704185719424

[illegible]

[illegible] SMMU enable [illegible] SMMU [illegible]

| [illegible] | [illegible] | | ——– | ————————————————————– | | StreamID | [illegible] SMMU [illegible] SMMU [illegible] Endpoint[illegible]SMMU [illegible] StreamID [illegible](SubstreamID [illegible] PCIe PASID [illegible]) | |STE|Stream Table Entry, STE [illegible] stage2 [illegible] CD [illegible]Context Descriptor[illegible].| |CD|Context Descriptor, [illegible] stage1 [illegible]|

[illegible]STE [illegible] CD[illegible]

STE[illegible]stream table entry[illegible] stage1 [illegible]CD [illegible]**CD** [illegible] **stage1** [illegible]stage2 [illegible] stage2 [illegible]

[illegible]

image-20240705110509333

image-20240705110521407

STE [illegible] 64B[illegible]SMMU [illegible] STRTAB _BASE [illegible] STE[illegible] STRTAB _BASE [illegible] STE [illegible] Stream id [illegible] STE [illegible] linear [illegible] STRTAB _BASE + sid * 64[illegible] STE [illegible] 64B[illegible] STE[illegible] 2-level [illegible] sid [illegible] L1 _STD[illegible]STRTAB _BASE + sid[9:8] * 8, [illegible] L1 _STD [illegible] 8B[illegible], L1 _STD [illegible] sid [illegible] STE[illegible]l2ptr + sid[7:0] * 64[illegible].

[illegible] STE [illegible] stream bypass, [illegible] stage1, stage2 [illegible] stage1 + stage2 [illegible]

image-20240705110739974

CD

image-20240705131054747

image-20240705131413291

[illegible] SubStreamID [illegible] CD [illegible]SubStreamID [illegible] SubStreamID[illegible]

SMMU command queue [illegible] event queue

[illegible] Command Queue [illegible] Event Queue [illegible] SMMU [illegible] 2 [illegible] Queue [illegible] Command queue [illegible] SMMU [illegible] command queue, SMMU [illegible] command queue [illegible] [illegible] Event Queue [illegible] SMMU [illegible] [illegible]SMMU [illegible] Event queue [illegible] Event queue [illegible]

streamID [illegible] substreamID

StreamID

StreamID is generated from the PCI RequesterID so that **StreamID[15:0] ==RequesterID[15:0]**

SubstreamID

The SubstreamID is equivalent to a **PCIe PASID**. Because the concept can be applied to non-PCIe systems, it has been given a more generic name in the SMMU. The maximum size of SubstreamID, 20 bits, matches the maximum size of a PCIe PASID.

c2k [illegible] StreamID N [illegible] 24[illegible] StreamID[23:0]

StreamID[15:0][illegible] pcie [illegible] RequesterID[15:0].

StreamID[23:16][illegible] Root Complex[illegible] c2000 [illegible] root complex [illegible] id [illegible] [illegible]:

SubstreamID [illegible] pcie [illegible] PASID

[illegible] SVA

[toc]

[illegible] SVA [illegible]

SVA (Shared Virtual Addressing) [illegible] SVA [illegible] SVM (Shared Virtual Memory)[illegible] SVA[illegible] Secure Virtual Machine [illegible]

image-20240705132416014

SVA [illegible]

- [illegible] CPU [illegible]GPU/NPU/SmartNIC/FPGA/ASIC/Accelerator [illegible]
- device [illegible] cpu [illegible]
- [illegible] DMA [illegible] cpu [illegible] pin DMA page

SVA [illegible]

1. [illegible] PASID
2. IO page fault [illegible] PCIe [illegible] PRI
3. **MMU [illegible] IOMMU [illegible]**

[illegible]

C2K [illegible] PPC [illegible] MMU[illegible] ARM [illegible] SMMU[illegible]

ARM SMMU [illegible]

PPC MMU [illegible]

SVA [illegible]MMU [illegible] SMMU [illegible] page table[illegible] SMMU [illegible] 64bit EA[illegible]MMU [illegible] 64bit EA [illegible] SLB [illegible] 78bit VA[illegible]

SVA [illegible] Intel [illegible] ARM [illegible]

image-20240705132753779

[illegible]device [illegible] BDF [illegible] streamID[illegible] BDF [illegible]CPU [illegible] process [illegible] PASID [illegible] DMA [illegible] process [illegible]

[illegible] PASID[illegible]

PASID

[illegible] PASID[illegible] PCIe spec [illegible] PASID TLP prefix [illegible] 20 bit [illegible] PASID value[illegible] **process** [illegible] Requeseter ID (BDF) [illegible] Requester ID [illegible] PASID [illegible] Requester ID [illegible] PASID[illegible] PASID prefix [illegible] EP [illegible] PASID [illegible] ATS[illegible]PRI [illegible] PASID TLP prefix [illegible]

image-20240705133326367

Intel IOMMU[illegible]VT-d[illegible]

image-20240705133458991

Nvidia UVA

image-20240705133550787

NoC [illegible]

CHI [illegible]

[toc]

CHI [illegible]

[illegible] SoC [illegible]

[illegible]

image-20240715093501311

Component

RN[illegible][illegible]

HN[illegible][illegible] cache [illegible] CMN700 [illegible]HNF [illegible] SLC[illegible]Combined PoS/PoC(ordering)[illegible]Snoop Filter

SN[illegible][illegible] HN [illegible]

image-20240715094002408

[illegible]

[illegible] PCIe [illegible]

image-20240715093341219

Protocol [illegible][illegible] transaction [illegible]

Network [illegible][illegible] transaction [illegible] packet[illegible] SAM [illegible] flit [illegible]

Link [illegible][illegible] Flit [illegible]4 [illegible] 6 [illegible] channel[illegible]

Protocol [illegible]

CHI [illegible] tansaction [illegible] read[illegible]write[illegible]snoop[illegible]atomic[illegible]snoop [illegible]

| Type | Transaction Type | Description | | —— | ————— | ————————————— | | Read | **ReadNoSnoop** | [illegible] | | Read | **ReadOnce** | [illegible] | | Read | **ReadShared** | [illegible] | | Read | **ReadClean** | [illegible] | | Read | **ReadNotSharedDirty** | [illegible] | | Read | **ReadUnique** | [illegible] | | Snoop | **CleanUnique** | [illegible] | | Snoop | **CleanShared** | [illegible] | | Snoop | **MakeUnique** | [illegible] | | Snoop | **Evict** | [illegible] | | Snoop | **StashOnce** | [illegible] | | Snoop | **StashUnique** | [illegible] | | Snoop | **DVM Operations** | [illegible] DVM [illegible] TLB [illegible] | | Write | **WriteNoSnoop** | [illegible] | | Write | **WriteUnique** | [illegible]

[illegible] | | Write | **WriteLineUnique** | [illegible] | | Write | **WriteBack** | [illegible] | | Write | **WriteClean** | [illegible] | | Write | **WriteEvictFull** | [illegible] | | Atomic | **Atomic Operations** | [illegible]-[illegible]-[illegible] |

[illegible] transaction flow

image-20240715100839260

1. RN-F sends a Read request to HN-F.
2. HN-F sends a Read request to SN-F

The ID field values in the Read request are based on where the data response is to be sent. Data can be sent to the Requester or to the HN-F.

1. SN-F sends a data response directly to RN-F.
2. RN-F sends CompAck to HN-F as the request is ReadShared and requires CompAck to complete the transaction.

Network [illegible]

[illegible] tansaction [illegible] SAM [illegible] packet[illegible] srcid[illegible]tgtid [illegible]

SAM [illegible] RNSAM, HNSAM

[illegible] power of 2 hashing[illegible] hieracical hashing [illegible]

Link [illegible]

4 [illegible] 6 [illegible] channel[illegible]

TX: RSP DATA REQ

RX: RSP DATA SNP

image-20240715094934665

[illegible]

[illegible] cache[illegible] cache [illegible]

Cache line states

image-20240715102852878

MESI

image-20240715101909675

MOESI

MESI [illegible] M [illegible] CPU [illegible] MOESI [illegible] Owned[illegible] O[illegible] M [illegible] M [illegible] O [illegible] MOESI [illegible] O [illegible] S [illegible] E[illegible] M [illegible] I [illegible] MESI [illegible] MESI [illegible] S [illegible] MOESI [illegible] O [illegible] O [illegible] MOESI [illegible] S [illegible] S [illegible] CPU [illegible] O [illegible] CPU [illegible] O [illegible]

image-20240715101930420

[illegible]

| | address | data | | ——- | ——- | —- | | DDR | 0x100 | 0xa | | Cache 1 | 0x100 | 0xa | | Cache 2 | 0x100 | 0xa |

core 2 [illegible] cache [illegible]

| | address | data | | ——- | ——- | ————— | | DDR | 0x100 | 0xa | | Cache 1 | 0x100 | 0xa | | Cache 2 | 0x100 | **0xa –> 0xb** |

[illegible]

[illegible] HN [illegible] directory[illegible] snoop[illegible] cache [illegible] ddr [illegible]

[illegible]

image-20240715104218584

CHI [illegible]-[illegible]

[illegible]snooping[illegible]directory-based[illegible]AMBA CHI[illegible]Coherent Hub Interface[illegible]directory-based[illegible]

[illegible]

- [illegible]
- [illegible]/[illegible]

[illegible]

- [illegible]
- [illegible]

[illegible]

[illegible] CHI [illegible]

- [illegible]
- [illegible]

[illegible]

[illegible]

[illegible]

- [illegible]/[illegible]
- [illegible]

[illegible]

- [illegible]
- [illegible]

[illegible]

- [illegible]
- [illegible]

[illegible]

- DVM

NoC

[toc]

[illegible] CPU [illegible]On-Chip Interconnect[illegible]

[illegible]

[illegible]

1. [illegible]Shared Bus[illegible] a.[illegible] b.[illegible] c.[illegible]
2. [illegible]Crossbar Switch[illegible] a.[illegible] b.[illegible] c.[illegible]
3. [illegible]Ring Interconnect[illegible] a.[illegible] b.[illegible] c.[illegible]
4. [illegible]Mesh[illegible]Torus[illegible]Mesh and Torus Interconnects[illegible] a.[illegible] b.[illegible] c.[illegible]
5. [illegible]Network-on-Chip, NoC[illegible] a.[illegible] b.[illegible] c.[illegible]

image-20240724100350615

[illegible] 1. [illegible]

[illegible]Intel Broadwell [illegible] ring [illegible] core[illegible] core [illegible] FIFO [illegible]

image-20240724100555990

[illegible]Intel [illegible] Skylake [illegible] Mesh [illegible]

image-20240724100623758

[illegible]

1. [illegible]——[illegible]
2. [Efficient Barrier Implementation on the POWER8 Processor]
3. [illegible]

CMN700

[toc]

CMN-700 [illegible] HNF/HNI/HNP [illegible]

[illegible] HN[illegible]Home Node[illegible] HN [illegible] SAM [illegible] HNF [illegible]HN [illegible]

HNF

[illegible] DRAM [illegible] HNF [illegible]

HNF[illegible]

1. **System Level Cache** [illegible] The System Level Cache (SLC) is a last-level cache. The SLC allocation policy is exclusive for data lines, except where sharing patterns are detected and pseudo-inclusive for code lines, as indicated by the RN-Fs. All code lines can be allocated into the SLC on the initial request. When MTE is enabled, SLC stores data and tags.
2. **Combined PoS/PoC** [illegible] The combined Point-of-Serialization/Point-of-Coherency (PoS/PoC) is responsible for the ordering of all memory requests sent to the HN-F. Ordering includes serialization of multiple outstanding requests and actions to the same line, and request ordering as required by the RN-F.
3. **Snoop Filter** [illegible] The Snoop Filter (SF) tracks cachelines that are present in the RN-Fs. It reduces snoop traffic in the system by favoring directed snoops over snoop broadcasts when possible. This approach substantially reduces the snoop response traffic that might otherwise be required.

[illegible] snoop [illegible] Snoop Filter[illegible]

Ans[illegible]HN [illegible] invalid [illegible]

HNI

The I/O coherent Home Node (HN-I) is a Home Node for all CHI transactions targeting AMBA subordinate devices

The HN-I acts as a proxy for all the RNs of CMN-700, converting CHI transactions to ACE5-Lite transactions. The HN-I includes support for the correct ordering of Arm device types.

The HN-I does not support caching of any data read from or written to the downstream ACE5-Lite I/O subordinate subsystem. Any cacheable request that is sent to the HN-I does not result in any snoops being sent to RN-Fs in the system. Instead, the request is converted to the appropriate ACE5-Lite read or write command and sent to the downstream ACE5-Lite subsystem.

If an RN-F caches data that is read from or written to the downstream ACE5-Lite I/O subordinate subsystem, coherency is not maintained. Any subsequent access to that data reads from or writes to the ACE5-Lite I/O subordinate subsystem directly, ignoring the cached data

There are HN-I types with extra functionality. These types include:

- HN-T HN-I that has a debug trace controller and DVM Node. CMN-700 can have zero or more HN-I and HN-T instances.
- HN-D HN-I that has a debug trace controller, DVM Node, and configuration subordinate. CMN-700 must have exactly one HN-D instance.
- HN-P HN-I that is optimized for peer-to-peer PCIe traffic.
- HN-V A device that includes an HN-I and DVM Node (DN).

HNP

The I/O coherent Home Node with PCIe optimization (HN-P) is a device that includes the HN-I functionality and dedicated trackers for PCIe peer-to-peer traffic. HN-P can only be used to connect to PCIe subordinates.

[illegible]	HNF	HNI	HNP
[illegible]	System Level Cache Combined PoS/PoC Snoop Filter	[illegible] CHI [illegible] ACE5-Lite	[illegible] HNI [illegible] PCIe P2P [illegible]
[illegible]	Yes	[illegible]	[illegible] PCIe [illegible]
[illegible]	[illegible]	[illegible] ACE5-Lite [illegible]	[illegible] PCIe [illegible]
[illegible]	NA	[illegible] AMBA [illegible]	[illegible] PCIe [illegible]
[illegible]	[illegible]	[illegible]	[illegible] PCIe [illegible]

NI-700

[illegible]

[illegible]

[toc]

[illegible]

[illegible] [illegible] [illegible] ARM [illegible] [illegible] [illegible] [illegible] ARMv8 [illegible] EL3[illegible] EL2 [illegible] EL1 [illegible]

ARMv7 [illegible]ARMv7 [illegible] 0x00000000 [illegible] [illegible] 4 [illegible] [illegible] [illegible] ARMv8 [illegible] [illegible] 128 [illegible] [illegible] 32 [illegible] [illegible] ARMv8 [illegible] 64 [illegible] 32 [illegible] [illegible] 64 [illegible]

ARMv8 [illegible] Vector Base Address Register[illegible] VBAR[illegible]

ARM M7 [illegible] Arm v7 [illegible]32 [illegible]

[illegible]

image-20240710140146524

```
Address       Exception Type
0x00000000    Reset
0x00000004    Undefined Instruction
0x00000008    Software Interrupt (SWI)
0x0000000C    Prefetch Abort
0x00000010    Data Abort
0x00000014    Reserved
0x00000018    IRQ
0x0000001C    FIQ
```

[illegible]

[illegible]

[illegible] I/O [illegible]

[illegible]

1. [illegible]
2. [illegible]IRQ[illegible]
3. [illegible]
4. [illegible] NVIC [illegible] GIC[illegible]
5. [illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]

[illegible]Exception[illegible]

[illegible]

[illegible]

1. [illegible]
2. [illegible]
3. [illegible]
4. [illegible]
5. [illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]

| [illegible] | [illegible]Interrupt[illegible] | [illegible]Exception[illegible] | | ——– | ————— | ———————- | | [illegible] | [illegible] | [illegible] | | [illegible] | [illegible] | [illegible] | | [illegible] | [illegible] | [illegible] | | [illegible] | [illegible] | [illegible] | | [illegible] | [illegible] | [illegible] | | [illegible] | [illegible] | [illegible] |

[illegible]

[illegible] ARM [illegible]

ARM64 [illegible]——nIRQ [illegible] nFIQ [illegible]

[illegible] [illegible] ARM64 [illegible] ARM [illegible] IRQ[illegible]Interrupt Request[illegible] FIQ[illegible]Fast Interrupt Request[illegible]

PSTATE [illegible] CPU [illegible] ù I[illegible] IRQ[illegible] ù F[illegible] FIQ[illegible]

image-20240710144034412

image-20240710143838604

[illegible] NVIC [illegible] GIC

image-20240710143621431

NVIC[illegible]Nested Vectored Interrupt Controller[illegible]

NVIC [illegible] Cortex-M [illegible]NVIC [illegible]

1. [illegible]- [illegible]

- [illegible]

1. [illegible]- NVIC [illegible] Cortex-M [illegible]

- [illegible]

1. [illegible]- [illegible] NVIC [illegible]
2. [illegible]- [illegible]VTOR[illegible]

- [illegible] 240 [illegible] Cortex-M [illegible]

GIC[illegible]Generic Interrupt Controller[illegible]

GIC [illegible] Cortex-A [illegible]GIC [illegible]

1. [illegible]- GIC [illegible]

- [illegible]Distributor[illegible] CPU [illegible]CPU Interface[illegible]

1. [illegible]- [illegible]

- [illegible]

1. [illegible]- [illegible] 1020 [illegible]

- GICv3 [illegible] GICv4 [illegible]

1. [illegible]- GIC [illegible]SoC[illegible]

- [illegible]

[illegible]

- **NVIC**[illegible]- [illegible] **Cortex-M** [illegible]
- [illegible]
- [illegible]
- **GIC**[illegible]- [illegible] **Cortex-A** [illegible]
- [illegible]
- [illegible]

ARM GIC

[toc]

GIC [illegible]

[illegible] ARM [illegible] ARM7 [illegible] ARM9[illegible] [illegible] [illegible] 64 [illegible] 32 [illegible] [illegible]Interrupt Enable Register[illegible] IER[illegible] [illegible]Interrupt Status Register[illegible]ISR[illegible]

[illegible] SoC [illegible]

- [illegible]
- [illegible] CPU [illegible] PCIe [illegible]
- [illegible]

[illegible] core[illegible] core [illegible] PCIe [illegible] 500 [illegible]

[illegible] ARM [illegible] GIC[illegible] V4[illegible] IP [illegible] GIC-700[illegible] GIC [illegible] 13.1 [illegible] GIC-V1 [illegible] GIC-390 [illegible] IP [illegible] ARM [illegible] IP[illegible]

image-20240710145328332

C2K [illegible] GIC V4.1.

[illegible]

image-20240710153531028

MSI [illegible]

image-20240710153547516

GIC-V3

GIC-V3 [illegible] PCIe [illegible] MSIX [illegible] PCIe [illegible] PCIe [illegible] GIC-V3 [illegible] CPU [illegible] 8[illegible] GIC-V3 [illegible] IP [illegible] GIC-500 [illegible] GIC-600[illegible]

GIC-V4

GICv4 [illegible] GICv3 [illegible]

1. **vPE[illegible]**

GICv4 [illegible]vPE[illegible]

- [illegible]GICv4 [illegible]vPE[illegible]Hypervisor[illegible]

1. [illegible]

GICv4 [illegible]Direct Injection[illegible]

- [illegible]GICv4 [illegible]

1. [illegible]

GICv4 [illegible]

- [illegible] **vPE** [illegible]GICv4 [illegible]vPE[illegible]

1. [illegible]

GICv4 [illegible]

- [illegible]**vLPI Control**[illegible]GICv4 [illegible]

1. [illegible]

GICv4 [illegible]

- [illegible]GICv4 [illegible]

1. [illegible]

GICv4 [illegible]

- [illegible]GICv4 [illegible]

[illegible]

[illegible] 4 [illegible] ù [illegible]inactive[illegible] ù [illegible]pending[illegible] CPU [illegible] ù [illegible]active[illegible] CPU [illegible] ù [illegible]active and pending[illegible] CPU [illegible]

[illegible] ù [illegible]edge-triggered[illegible] ù [illegible]level-triggered[illegible]

GIC [illegible]

[illegible] GIC [illegible] GIC [illegible]

image-20240710151348491

[illegible] SoC [illegible] GIC-V2 [illegible]

- SGI [illegible]SGI [illegible] Linux [illegible]Inter-Processor Interrupt[illegible] IPI[illegible] CPU [illegible]
- PPI [illegible]PPI [illegible] CPU [illegible] CPU [illegible]local timer[illegible]
- SPI [illegible]

SGI [illegible] PPI [illegible] CPU [illegible] SPI [illegible] CPU [illegible]

GIC [illegible]distributor[illegible] CPU [illegible]

image-20240710152635739

GIC-V2 [illegible] CPU [illegible]CPU interface[illegible] 13.2 [illegible] CPU [illegible] CPU [illegible] CPU [illegible] CPU [illegible] nIRQ [illegible] nFIQ [illegible] CPU [illegible]

GIC-700

GIC-700 □□□□□

image-20240710152546004

gic-v3 □□□□□□□□□ Distributor □Redistributor □CPU interface □ ITS □Interrupt Translation Service□□

□□□

- Distributor □□□□ SPI □□□□priority□affinity□trigger-type□□□□□ sgi □spi □□□□ Redistributor □
- Redistributor □□□□ PPI □LPI □□□□priority□affinity□trigger-type□□□□ PPI □LPI □□ CPU interface□
- CPU interface □□□□□□ Redistributor □□□□□□□□□□□□□□□□ CPU□□□□ CPU □□□□□□□□ Distributor □ Redistributor □□□□□□□□□□□□□
- ITS □□□□□□□□□□□□□ msi □□□□□ msi □□□□□ LPI □□□□□□ Redistributor □

| □□□□ | □□□□ | □□ | | ——– | ——————————————— | ————– | | GIC | General Interrupt Controller | □□□ □□□□ | | SGI | Software Generated Interrupt | □□□□ | | PPI | Private Peripheral Interrupt | □□□□ | | SPI | Shared Peripheral Interrupt | □□□□ | | LPI | Locality-specific Peripheral Interrupt | □□□□ | | PE | Process Element | □□□□□ | | ITS | Interrupt Translation Service | □□□□□□ |

| □□ | □□□□ | | —- | ————————————————————— | | SGI | ① □□□□□□□□ ② □□ SGI □□□□□□□□□□□□□ security □□□□□ group□□□□□□□ PE□□□□□□ PE / □□ PE□ | | PPI | ① □□□□□□□□ ② □□ PPI □□□□□□□□□□□□□□□ | | SPI | ① □□□□□□□□ ② □□□□□□□□□□ ③ □□□□□□□□□□□□ PE□□□ PE / □□ PE□ | | LPI | ① □□ LPI □□□□ □forwarding □□/ITS □□□□□□□□□□□ ② □□□□□□□□ |

□□□□ linux kernel 5.10□

| □□□□ | □□□ | □□ | | ———- | ———– | ————————— | | SGI | 0 - 15 | □□ PE □□□□ | | PPI | 16 – 31 | □□ PE □□ □□ | | SPI | 32 - 1019 | | | □□□□□ | 1020 - 1023 | | | □□□□□ | 1024 - 1055 | | | □□ PPI | 1056 –1119 | GICv3.1 □□□□ □□ | | □□□□□ | 1120 - 4095 | | | □□ SPI | 4096 - 5119 | GICv3.1 □□□□□□ | | □□□□□ | 5120 - 8191 | | | LPI | 8192 - | □□□□□□□□□□□□□ |

□□□□ LPI □□□□□□ 8192 □□□□□□□□□□□ LPI □□□□□□□□□□□□□□ GICD _TYPER □□□ IDbits field □□□□□□□□□□□ □□ LPI□

image-20240710153719002

image-20240710153330434

□□□□□

GCI□□ GIC Cluster Interface□□□□□□ cluster □□□ Redistributor□□ c2000 □□□□ EXG □□□□ cluster□□□□□ core□8 □ thread□□□□□ GCI □□□□□ 8 □ Redistributor

c2k □□□□ ICI □□□□□□□□□□ ARM □ CPU interface □□

SGI □□

□□□□□□ core A □□□□□□ CPU interface□□ GICD □□□□□□□□□ threadde GICR □□□□□ interface □□□□□ thread□

PPI [illegible]

[illegible]PPI [illegible] GICR[illegible] interface [illegible] thread[illegible]

[illegible] EXG [illegible] timer[illegible]

SPI [illegible]

[illegible]SPI [illegible]GICD [illegible] GICR[illegible]GICR [illegible] interface [illegible] thread[illegible]

[illegible] PCIe [illegible]

LPI [illegible]

[illegible]MSI [illegible]ITS [illegible] GICR[illegible]GICR [illegible] interface [illegible] thread[illegible]

image-20240710160158086

[illegible]

1. [illegible] ITS[illegible] device id [illegible] event id[illegible]device id [illegible]event id [illegible]
2. ITS [illegible] GICR [illegible] CPU[illegible]

 a) Device table [illegible] device [illegible] ITT [illegible]

 b) ITT [illegible] event id [illegible] collection id [illegible]

 c) Collection table [illegible] collection id [illegible] GICR [illegible]

1. ITS [illegible] ITS [illegible] ITS [illegible] CMDQ [illegible]

[illegible] c2k [illegible] MSI [illegible] its->gicd->gicr [illegible] arm [illegible]its [illegible] redistributor[illegible]GICR[illegible] distributor(GICD)[illegible]

[illegible]GICD [illegible] ITS [illegible] GICD[illegible]GICD [illegible] ddr [illegible] GICR[illegible]c2k [illegible] arm [illegible]

X86 APIC

RISCV AIA

[toc]

SoC

Cortex M3 MCU

[toc]

M3

image-20240710142634874

MCU [illegible]

image-20240709153614160

image-20240709153633986

[illegible]

image-20240709154314824

image-20240709154051664

image-20240709154708302

ARM N2 [illegible]

ARM N2 [illegible]ARM Neoverse N2 reference design[illegible] ARM [illegible]Neoverse N2 [illegible] ARM [illegible] CPU [illegible]

1. [illegible]N2 [illegible]HPC[illegible]
2. [illegible]Neoverse N2 [illegible]
3. [illegible]ARM N2 [illegible]
4. [illegible]N2 [illegible]
5. [illegible]Neoverse N2 [illegible] ARM [illegible] ARM [illegible]

[illegible]ARM N2 [illegible]

ARM Neoverse N2 [illegible] ARM [illegible]IP[illegible] IP [illegible]

1. **ARM Neoverse N2 CPU** [illegible] N2 [illegible]N2 [illegible] ARMv9 [illegible]
2. **System Level Cache (SLC)**[illegible]ARM N2 [illegible]SLC[illegible]
3. **CCIX** [illegible] **CXL** [illegible]N2 [illegible] CCIX[illegible]Cache Coherent Interconnect for Accelerators[illegible] CXL[illegible]Compute Express Link[illegible]
4. **AMBA (Advanced Microcontroller Bus Architecture)** [illegible]AMBA [illegible]SoC[illegible] IP [illegible]N2 [illegible] AMBA [illegible]
5. **TrustZone** [illegible]ARM TrustZone [illegible]
6. **Mali GPU**[illegible] N2 [illegible] CPU [illegible] ARM [illegible] Mali GPU[illegible]
7. **SMMU (System Memory Management Unit)**[illegible]

[illegible] IP [illegible]ARM Neoverse N2 [illegible]

[illegible] High Level [illegible]

[illegible] Arm Neoverse N2 reference design Technical Overview

image-20240704104921533

[illegible] CMN-700[illegible] PICe [illegible] SMMU [illegible] NI-700[illegible] RNI [illegible] SCP[illegible] GIC [illegible] SNF [illegible] memory controller[illegible] core[illegible]

□□

ARM □□□□□□

1. □□□□□□□□□□BMC □□□□□□□ C2000 CHIP □ POR _N□pervsoc □□□□□□□□□□
2. □ otprom valid □□□□□□□ SCP SRAM □ BISR □□□□PVO(SCP)□□□ ROM □otprom □□□□□□□□□□
3. □ SCP □□□□□□ SRAM□
4. SCP ROM FW □ GPIO □□□□□□ CHIP ID,□□ CHIP ID □□□□□□□□ CHIP;
5. □□□□□□□□□□□ I2C/FSI□□□□□□□□□□□□□□□
6. □ CHIP □ SCP □□□□□□ QSPI;
7. □ CHIP □ SCP □□□□□ CHIP □□□□□□□□□□□□□ CHIP □□□
8. □ CHIP □ SCP □□□□□ CHIP SCP □□□□□□□
9. □ CHIP □ SCP □ QSPI □□□□□ SCP RAM FW □□ CHIP □ SRAM;
10. □ CHIP □ SCP □□□□ 8 □□□□□□□□□□ SCP RAM FW □□□□□□□□□□ SCP,□ SCP □□□□□□□ SRAM:
11. □ CHIP □ SCP □□□□□ SRAM,□□ SCP RAM FW:
12. □ CHIP □ SCP □□ QSPI □□ C2C □□□□□□□□□□□□□□□□ SCP□
13. □ CHIP □ SCP □□ CHIP □ SCP □□□□□ C2C □□□□□□ C2C;
14. □ SCP □□ CHIP ID □□□ CMN□OCM(SLC)□ SAM;
15. □ CHIP □ SCP □□□□□ MCP RAM FW,□□ MCP(□□□)□+
16. □ SCP □□□□□ CHIP □□□□ L2□NCU□SCOM/FIR;+
17. □ CHIP □ SCP □□□□□ Hostboot base □□□□ CHIP □ OCM(SLC);+
18. □ CHIP □ SCP □□□ CHIP □□□□□□□□□□□□□□□□□
19. □□□□□□□□□□□□ srest□0x100□□□□□

□ core-hostboot

1. □□ kernel other thread spinlock,□□□□□□□□ thread □□□□
2. □□ hostboot kernel □□□□□□□□□ C/C++□□ main □□□+
3. □□ hostboot kernel □□□□□□□□□□□
4. □□□ MMU(SLB/TLB),□□ DIMM □□□□□□□□ OCM(SLC,□□□□□□ 28 □* 2MB=56MB□□□□□ 3/4□□
5. □□□□□□□□ TB □□□ CPU □□□
6. □ Scratch □□□□□□ FSP/BMC;
7. □□ CPU □□ init main □□□
8. □□ kernel other thread spinlock □ 1□□□□□ thread □□□□

9. hostboot [illegible] task [illegible] task [illegible] user [illegible] kernel [illegible] hypersior [illegible]
10. init main [illegible] VFS;
11. VFS [illegible] base modules[illegible]

[illegible] init [illegible]

1. [illegible] initservice [illegible]
2. initservice [illegible] g _taskinfolist [illegible]
3. secureboot _base/ipmibase/pnor/vfs/extinitsvc [illegible]
4. [illegible] hostboot extended image [illegible]
5. extinitsvc [illegible] g _exttaskinfolist [illegible]
6. [illegible] istepdisp[illegible] hostboot [illegible]
7. [illegible] istep [illegible] hostboot [illegible]
8. istep 21.03 [illegible] payload(skiboot,OPAL);+
9. [illegible] payload(skiboot,OPAL)[illegible]

[illegible] core-skiboot

1. [illegible]C2000 [illegible] skiboot [illegible]
2. [illegible] thread [illegible] skiboot [illegible] relocate [illegible] thread [illegible]
3. [illegible] thread [illegible] boot flag=1,[illegible] thread [illegible]
4. [illegible] thread [illegible] main cpu entry,[illegible] thread [illegible] secondary cpu entry;
5. [illegible] uart(console);
6. [illegible] skiboot [illegible]/[illegible]
7. [illegible] skiboot [illegible]
8. [illegible] opal call table;
9. [illegible] hostboot [illegible] device tree [illegible] HDAT;
10. [illegible] DT [illegible] SCOM/eSPI(LPC):
11. [illegible] DT [illegible]C2000 [illegible] DIMM [illegible]
12. [illegible] DT [illegible] homer[illegible]C2000 [illegible]
13. [illegible] DT [illegible] CPU(thread);
14. [illegible] opal [illegible] DT;
15. [illegible]
16. [illegible] BMC [illegible] ipmi/BT/UART;
17. [illegible]GIC[illegible]

18.[illegible] CPU[illegible]thread[illegible]

19.[illegible] sreset[illegible] CPU [illegible]

20.[illegible] TOD[illegible] TOD [illegible]

21.[illegible] sensor [illegible] OPAL [illegible]

22.[illegible]

23.NV [illegible]

24.[illegible]jiayu[illegible]

25.[illegible]TPM[illegible]

26.[illegible] opal console(UART)[illegible]

1. PCIE HB [illegible] PCIE [illegible] (RC [illegible]CXL [illegible] PCIe [illegible]device tree [illegible])

28.PCIE SLOT [illegible]

29.[illegible]

30.[illegible]

31.[illegible] zImage(boot loader [illegible] petitboot);

32.[illegible] boot loader [illegible]

image-20240704111929075

image-20240704111945789

CMN OCM(on chip memory) [illegible] L3 cache[illegible] DDR [illegible] Intel [illegible] cache as ram [illegible]

[illegible] OCM [illegible] memory [illegible]CMN OCM(on chip memory) [illegible] 28*2=56M [illegible] 3/4[illegible] 32M[illegible]

[illegible]

SCP ram [illegible] DDR training[illegible] OCM [illegible] flash [illegible] DDR [illegible]

[illegible] OCM [illegible] SNF [illegible]

[illegible]

SCP [illegible] OCM [illegible]SCP RAM [illegible] hostboot [illegible] OCM [illegible]skiboot [illegible] OCM [illegible] skiboot [illegible] ddr training[illegible] DDR[illegible] OCM[illegible] OCM[illegible] cache [illegible]

hostboot [illegible] skiboot [illegible] 0x800 0000 [illegible]

32M [illegible] 0x200 0000, [illegible] skiboot [illegible] 0x3000 0000 [illegible] OCM[illegible] 0x0[illegible] 0x0 [illegible]

RISC-V ☒☒☒☒☒☒

☒☒

Firmware ☒☒☒☒☒☒☒

image-20240711102511545

ARM64 vs RISC-V

image-20240711103010418

uboot ☒☒☒☒☒☒ EDK2☒☒☒☒ PEI ☒☒☒☒☒☒☒ DXE ☒☒☒☒☒

image-20240711103301123

☒☒☒☒☒☒☒

ZSBL: SCP ROM

FSBL: SCP RAM

image-20240711105826248

Cache as RAM & On chip memory

☒☒

☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

x86 ☒☒☒ Cache as RAM, CAR;

ARM ☒☒☒☒ On Chip Memory☒OCM☒

☒☒☒☒☒☒☒☒☒☒ DDR ☒☒☒☒☒☒☒☒☒☒☒ cache ☒☒ RAM☒☒☒☒ C ☒☒☒☒☒

CAR

EDK2 System Table

image-20240729101921741

☒ `MdeModulePkg\Core\Dxe\DxeMain\DxeMain.c` ☒☒☒☒

```
  //
  // Allocate the EFI System Table and EFI Runtime Service Table from
EfiRuntimeServicesData
  // Use the templates to initialize the contents of the EFI System Table and EFI
Runtime Services Table
  //
  gDxeCoreST = AllocateRuntimeCopyPool (sizeof (EFI_SYSTEM_TABLE),
&mEfiSystemTableTemplate);
  ASSERT (gDxeCoreST != NULL);

  gDxeCoreRT = AllocateRuntimeCopyPool (sizeof (EFI_RUNTIME_SERVICES),
&mEfiRuntimeServicesTableTemplate);
  ASSERT (gDxeCoreRT != NULL);

  gDxeCoreST->RuntimeServices = gDxeCoreRT;
```

☒☒☒

```
///
/// EFI System Table
///
typedef struct {
  ///
  /// The table header for the EFI System Table.
  ///
  EFI_TABLE_HEADER                  Hdr;
  ///
  /// A pointer to a null terminated string that identifies the vendor
  /// that produces the system firmware for the platform.
  ///
  CHAR16                            *FirmwareVendor;
  ///
  /// A firmware vendor specific value that identifies the revision
  /// of the system firmware for the platform.
  ///
  UINT32                            FirmwareRevision;
  ///
  /// The handle for the active console input device. This handle must support
  /// EFI_SIMPLE_TEXT_INPUT_PROTOCOL and EFI_SIMPLE_TEXT_INPUT_EX_PROTOCOL.
  ///
  EFI_HANDLE                        ConsoleInHandle;
  ///
  /// A pointer to the EFI_SIMPLE_TEXT_INPUT_PROTOCOL interface that is
  /// associated with ConsoleInHandle.
  ///
  EFI_SIMPLE_TEXT_INPUT_PROTOCOL    *ConIn;
  ///
  /// The handle for the active console output device.
  ///
  EFI_HANDLE                        ConsoleOutHandle;
  ///
  /// A pointer to the EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL interface
  /// that is associated with ConsoleOutHandle.
  ///
  EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL   *ConOut;
  ///
  /// The handle for the active standard error console device.
  /// This handle must support the EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL.
  ///
  EFI_HANDLE                        StandardErrorHandle;
  ///
  /// A pointer to the EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL interface
  /// that is associated with StandardErrorHandle.
  ///
  EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL   *StdErr;
  ///
  /// A pointer to the EFI Runtime Services Table.
  ///
  EFI_RUNTIME_SERVICES              *RuntimeServices;
  ///
  /// A pointer to the EFI Boot Services Table.
  ///
  EFI_BOOT_SERVICES                 *BootServices;
  ///
  /// The number of system configuration tables in the buffer ConfigurationTable.
  ///
  UINTN                             NumberOfTableEntries;
  ///
  /// A pointer to the system configuration tables.
  /// The number of entries in the table is NumberOfTableEntries.
  ///
  EFI_CONFIGURATION_TABLE           *ConfigurationTable;
} EFI_SYSTEM_TABLE;
```

EDK2 Event

[toc]

□□

□□

□□□□□□□□□UEFI □□□□□□□□□□□□□□□□□□□□□□□

□□ □□□/□□□ □□□□□□ intel □□□ 8253/8254 □□□□□□□□□□ PC □□□□□□□□ IC□□□□□□□□□□□□□□□□□□□□□□□□ MSC8051,□□□□□□□□□□□□

□□□□ Timer □□□ cpu □□

□ PC/AT □□ UEFI □□□□□□□□□ 8254 □□□□□□ Timer□□□□□ mode 3□□□□□□ 1.1931816MHz□□□□ tick □mTimerPeriod□□□□□□ 1ms□10000 □ tick □□□□□=10000*100ns□□□□□ CPU □□ IRQ0□□□□□□□ 0 □□□□□□ □ OUT pin □ 8259 □ IRQ0 □□□□□□□□ CPU □□□□□□□□□□□ Timer □□□□□□□

□□□□□□□□□□□□□□ cpu archprotocol □ Legacy8259Protocol□□□□□□□□□,□□□□□□□□cpu □□ IDT □□□□□□□□ □ Timer □□□□ TimerInterruptHandler□

Timer □□

□□□□ Timer □□□□□□□□□□□□□□□□□□□□□□□□□□□□

1.□□□

2.□□□□

3.□□□□□□□□□□□□□□□□□□□□□□□

4.□□□□□□□ Timer □□□□□□□□□ CoreTimerTick(mTimerPeriod)/CoreTimerTick(100ms)□□□□ TimerArchprotocol □ install □□□□ EVT _NOTIFY _SIGNAL □□□ Event □□□□□□

5.□□ Sytem time □□ mEfiSystemTime =mEfiSystemTime+ 1Tick(100ms)

6. □ timer event list □□□□□□□□□□□□□□ event□□□□□□□□ BS->SignalEvent □ signal □□□

BS->SignalEvent—>CoreNotifyEvent–>CoreRestoreTpl–>CoreDispatchEventNotifies–>A.□□□ EVT _NOTIFY _SIGNAL □□□□□ count□Event->SignalCount = 0□□□□□□□□□□ Event->NotifyFunction (Event, Event->NotifyContext)□□□□□□□□□□□□□□□ Tpl □ gEventPending mask bit□□□□ event□

7.□□□□□□□□□□□□□□□□□□□□□□□

8.□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□

1. gEventQueue— □□□□□□A list of event's to notify for each priority level□□□□□□□□□

2. mEfiTimerList—-□□□□□□□□□□ timer □□□ event □□□

3. gEventSignalQueue —[illegible] A list of events to signal based on EventGroup type

4. [illegible] EVT _NOTIFY _WAIT [illegible] event[illegible] BS-> **CoreCheckEvent**->CoreNotifyEvent–>CoreRestoreTpl–>CoreDispatchEventNotifies [illegible] gEventQueue [illegible] IEvent [illegible]

gEventPending [illegible] gEventQueue [illegible] 1[illegible] 0[illegible]

gEventPending [illegible] 32 [illegible] 1 [illegible]

[illegible] x86 [illegible]IRQ[illegible]

- IRQ0: [illegible]
- IRQ1: [illegible]
- IRQ2: [illegible]
- IRQ3: [illegible] 2[illegible]COM2[illegible]
- IRQ4: [illegible] 1[illegible]COM1[illegible]
- IRQ5: [illegible]
- IRQ6: [illegible]
- IRQ7: [illegible] 1[illegible]LPT1[illegible]
- [illegible] IRQ [illegible]

event [illegible]

```
//
// Task priority level
//
#define TPL_APPLICATION       4
#define TPL_CALLBACK          8
#define TPL_NOTIFY            16
#define TPL_HIGH_LEVEL        31
```

31 [illegible]

[illegible]

image-20240716144753316

[illegible] DXE main [illegible]

```
  //
  // Initialize the Event Services
  //
  Status = CoreInitializeEventServices ();

...

  //
  // Register for the GUIDs of the Architectural Protocols, so the rest of the
  // EFI Boot Services and EFI Runtime Services tables can be filled in.
  // Also register for the GUIDs of optional protocols.
  //
  CoreNotifyOnProtocolInstallation ();
```

[illegible] event [illegible]

[illegible]

```
/**
  Called by the platform code to process a tick.

  @param  Duration               The number of 100ns elapsed since the last call
                                 to TimerTick

**/
VOID
EFIAPI
CoreTimerTick (
  IN UINT64   Duration
  )
{
  IEVENT          *Event;

  //
  // Check runtiem flag in case there are ticks while exiting boot services
  //
  CoreAcquireLock (&mEfiSystemTimeLock);

  //
  // Update the system time
  //
  mEfiSystemTime += Duration;

  //
  // If the head of the list is expired, fire the timer event
  // to process it
  //
  if (!IsListEmpty (&mEfiTimerList)) {
    Event = CR (mEfiTimerList.ForwardLink, IEVENT, Timer.Link, EVENT_SIGNATURE);

    if (Event->Timer.TriggerTime <= mEfiSystemTime) {
      CoreSignalEvent (mEfiCheckTimerEvent);
    }
  }

  CoreReleaseLock (&mEfiSystemTimeLock);
}
```

[illegible] timer [illegible] event[illegible]

```
/**
  Initializes "event" support.

  @retval EFI_SUCCESS            Always return success

**/
EFI_STATUS
CoreInitializeEventServices (
  VOID
  )
{
  UINTN        Index;

  for (Index=0; Index <= TPL_HIGH_LEVEL; Index++) {
    InitializeListHead (&gEventQueue[Index]);
  }

  CoreInitializeTimer ();

  CoreCreateEventEx (
    EVT_NOTIFY_SIGNAL,
    TPL_NOTIFY,
    EfiEventEmptyFunction,
    NULL,
    &gIdleLoopEventGuid,
    &gIdleLoopEvent
    );

  return EFI_SUCCESS;
}
```

- [illegible] gEventQueue— [illegible]A list of event's to notify for each priority level
- CoreInitializeTimer [illegible]

```
/**
  Initializes timer support.

**/
VOID
CoreInitializeTimer (
  VOID
  )
{
  EFI_STATUS  Status;

  Status = CoreCreateEventInternal (
             EVT_NOTIFY_SIGNAL,
             TPL_HIGH_LEVEL - 1,
             CoreCheckTimers,
             NULL,
             NULL,
             &mEfiCheckTimerEvent
             );
  ASSERT_EFI_ERROR (Status);
}
```

⊠⊠⊠⊠⊠⊠⊠⊠

```
#include <Uefi.h>
#include <Library/UefiBootServicesTableLib.h>
#include <Library/UefiLib.h>

EFI_EVENT TimerEvent;
EFI_EVENT ExitBootServicesEvent;
BOOLEAN TimerTriggered = FALSE;

VOID EFIAPI TimerEventHandler(
    IN EFI_EVENT Event,
    IN VOID *Context
) {
    TimerTriggered = TRUE;
    Print(L"Timer event triggered!\n");
}

EFI_STATUS
EFIAPI
UefiMain(
    IN EFI_HANDLE ImageHandle,
    IN EFI_SYSTEM_TABLE *SystemTable
) {
    EFI_STATUS Status;

    // □□□□□□□□□□□□□□10000000 100ns (1□)□□□
    Status = gBS->CreateEvent(
        EVT_TIMER | EVT_NOTIFY_SIGNAL,
        TPL_CALLBACK,
        TimerEventHandler,
        NULL,
        &TimerEvent
    );
    if (EFI_ERROR(Status)) {
        Print(L"Failed to create event: %r\n", Status);
        return Status;
    }
    // □□□□□□1 □□□□
    Status = gBS->SetTimer(
        TimerEvent,
        TimerRelative,
        10000000 // 100ns □□□10000000 = 1 □
    );
    if (EFI_ERROR(Status)) {
        Print(L"Failed to set timer: %r\n", Status);
        gBS->CloseEvent(TimerEvent);
        return Status;
    }

    // □□□□□□
    while (!TimerTriggered) {
        gBS->Stall(100000); // □□ 100ms
    }

    // □□
    gBS->CloseEvent(TimerEvent);

    return EFI_SUCCESS;
}
```

⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠1 ⊠⊠⊠⊠⊠

⊠⊠⊠⊠⊠ gBS->CreateEvent ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

```
typedef struct {
    UINT32 Type;
    EFI_TPL NotifyTpl;
    EFI_EVENT_NOTIFY NotifyFunction;
    VOID *NotifyContext;
    BOOLEAN Triggered;
} EFI_EVENT_INTERNAL;

EFI_STATUS
EFIAPI
CreateEvent (
    IN UINT32 Type,
    IN EFI_TPL NotifyTpl,
    IN EFI_EVENT_NOTIFY NotifyFunction,
    IN VOID *NotifyContext,
    OUT EFI_EVENT *Event
) {
    EFI_STATUS Status;
    EFI_EVENT_INTERNAL *NewEvent;

    // □□□□
    if (Event == NULL || (Type & EVT_NOTIFY_SIGNAL && NotifyFunction == NULL)) {
        return EFI_INVALID_PARAMETER;
    }

    // [illegible]
    NewEvent = AllocateZeroPool(sizeof(EFI_EVENT_INTERNAL));
    if (NewEvent == NULL) {
        return EFI_OUT_OF_RESOURCES;
    }

    // [illegible]
    NewEvent->Type = Type;
    NewEvent->NotifyTpl = NotifyTpl;
    NewEvent->NotifyFunction = NotifyFunction;
    NewEvent->NotifyContext = NotifyContext;
    NewEvent->Triggered = FALSE;

    // [illegible]
    Status = RegisterEvent(NewEvent);
    if (EFI_ERROR(Status)) {
        FreePool(NewEvent);
        return Status;
    }

    // [illegible]
    *Event = (EFI_EVENT)NewEvent;

    return EFI_SUCCESS;
}
```

[illegible] event [illegible]

[illegible] event [illegible]

```
  if ((Type & EVT_NOTIFY_SIGNAL) != 0x00000000) {
    //
    // The Event's NotifyFunction must be queued whenever the event is signaled
    //
    InsertHeadList (&gEventSignalQueue, &IEvent->SignalLink);
  }
```

[illegible] 32 [illegible] bitmap [illegible] 1[illegible]

[illegible]

SignalEvent [illegible] event [illegible] 32 [illegible]

```
/**
  Signals the event.  Queues the event to be notified if needed.
  @param  UserEvent              The event to signal .

  @retval EFI_INVALID_PARAMETER  Parameters are not valid.
  @retval EFI_SUCCESS            The event was signaled.

**/
EFI_STATUS
EFIAPI
CoreSignalEvent (
  IN EFI_EVENT    UserEvent
  )
/**
  Queues the event's notification function to fire.

  @param  Event                  The Event to notify

**/
VOID
CoreNotifyEvent (
  IN  IEVENT      *Event
  )
{

  //
  // Event database must be locked
  //
  ASSERT_LOCKED (&gEventQueueLock);

  //
  // If the event is queued somewhere, remove it
  //

  if (Event->NotifyLink.ForwardLink != NULL) {
    RemoveEntryList (&Event->NotifyLink);
    Event->NotifyLink.ForwardLink = NULL;
  }

  //
  // Queue the event to the pending notification list
  //

  InsertTailList (&gEventQueue[Event->NotifyTpl], &Event->NotifyLink);
  gEventPending |= (UINTN)(1 << Event->NotifyTpl);
}
```

[illegible]

Duang[illegible]CoreTimerTick [illegible] TPL[illegible] restore TPL[illegible]

[illegible] timer [illegible] event[illegible]mEfiTimerList[illegible]

[illegible] TPL [illegible]

```
 while (gEventPending != 0)
{
  PendingTpl = (UINTN) HighBitSet64 (gEventPending);
  if (PendingTpl <= NewTpl) {
   break;
  }
  gEfiCurrentTpl = PendingTpl;
  if (gEfiCurrentTpl < TPL_HIGH_LEVEL) {
   CoreSetInterruptState (TRUE);
  }
  CoreDispatchEventNotifies (gEfiCurrentTpl);
  //创创 gEventQueue [gEfiCurrentTpl]创创创创创 event 创 Notification 创创
 }
```

[illegible]gEventPending [illegible] 32 [illegible] 1 [illegible]

[illegible]

1. [illegible]
2. [illegible] TPL [illegible] TPL [illegible] 16[illegible] 17[illegible] 16 [illegible] TPL [illegible] 17 [illegible]
3. [illegible] pending [illegible] pending [illegible] TPL [illegible] 16[illegible] 15[illegible] 16 [illegible]

[illegible]——[illegible]

[illegible] waitforevent [illegible] 4[illegible] 10ms [illegible] 31[illegible] RestoreTpl [illegible] 4, [illegible] 31 [illegible] 4 [illegible] 30 [illegible] 5 [illegible] 4 [illegible]

CoreTimerTick [illegible]

```
/**
 Lowers the task priority to the previous value.   If the new
 priority unmasks events at a higher priority, they are dispatched.

 @param  NewTpl  New, lower, task priority

**/
VOID
EFIAPI
CoreRestoreTpl (
 IN EFI_TPL NewTpl
 )
```

[illegible]

```
{
  EFI_TPL     OldTpl;
  EFI_TPL     PendingTpl;
```

- OldTpl: [illegible]
- PendingTpl: [illegible]

[illegible]

```
  OldTpl = gEfiCurrentTpl;
  if (NewTpl > OldTpl) {
    DEBUG ((EFI_D_ERROR, "FATAL ERROR - RestoreTpl with NewTpl(0x%x) > OldTpl(0x%x)\n",
NewTpl, OldTpl));
    ASSERT (FALSE);
  }
  ASSERT (VALID_TPL (NewTpl));
```

- OldTpl = gEfiCurrentTpl: [illegible]
- [illegible] NewTpl [illegible] NewTpl [illegible] OldTpl[illegible]

[illegible]

```
  if (OldTpl >= TPL_HIGH_LEVEL  &&  NewTpl < TPL_HIGH_LEVEL) {
    gEfiCurrentTpl = TPL_HIGH_LEVEL;
  }
```

- [illegible]TPL_HIGH_LEVEL[illegible] TPL_HIGH_LEVEL[illegible] TPL_HIGH_LEVEL[illegible]

[illegible]

```
  while (gEventPending != 0) {
    PendingTpl = (UINTN) HighBitSet64 (gEventPending);
    if (PendingTpl <= NewTpl) {
      break;
    }

    gEfiCurrentTpl = PendingTpl;
    if (gEfiCurrentTpl < TPL_HIGH_LEVEL) {
      CoreSetInterruptState (TRUE);
    }
    CoreDispatchEventNotifies (gEfiCurrentTpl);
  }
```

- while (gEventPending != 0): [illegible]
- PendingTpl = (UINTN) HighBitSet64 (gEventPending): [illegible]
- [illegible] PendingTpl [illegible] NewTpl[illegible]
- [illegible] PendingTpl[illegible] TPL_HIGH_LEVEL [illegible]
- [illegible] CoreDispatchEventNotifies [illegible]

[illegible]

```
  gEfiCurrentTpl = NewTpl;

  if (gEfiCurrentTpl < TPL_HIGH_LEVEL) {
    CoreSetInterruptState (TRUE);
  }

}
```

- [illegible] NewTpl[illegible]
- [illegible] TPL_HIGH_LEVEL[illegible]

CoreDispatchEventNotifies [illegible]

```
/**
  Dispatches all pending events.

  @param  Priority               The task priority level of event notifications
                                 to dispatch

**/
VOID
CoreDispatchEventNotifies (
  IN EFI_TPL      Priority
  )
```

[illegible]

```
{
  IEVENT          *Event;
  LIST_ENTRY      *Head;
```

- Event: [illegible]
- Head: [illegible]

[illegible]

```
  CoreAcquireEventLock ();
  ASSERT (gEventQueueLock.OwnerTpl == Priority);
  Head = &gEventQueue[Priority];
```

- CoreAcquireEventLock ()：[illegible]
- ASSERT (gEventQueueLock.OwnerTpl == Priority)：[illegible]
- Head = &gEventQueue[Priority]：[illegible]

[illegible]

```
  while (!IsListEmpty (Head)) {

    Event = CR (Head->ForwardLink, IEVENT, NotifyLink, EVENT_SIGNATURE);
    RemoveEntryList (&Event->NotifyLink);

    Event->NotifyLink.ForwardLink = NULL;
```

- while (!IsListEmpty (Head))：[illegible]
- Event = CR (Head->ForwardLink, IEVENT, NotifyLink, EVENT_SIGNATURE)：[illegible]
- RemoveEntryList (&Event->NotifyLink)：[illegible]
- Event->NotifyLink.ForwardLink = NULL：[illegible]

[illegible] SIGNAL [illegible]

```
    if ((Event->Type & EVT_NOTIFY_SIGNAL) != 0) {
      Event->SignalCount = 0;
    }
```

- [illegible] EVT_NOTIFY_SIGNAL[illegible] SignalCount[illegible]

[illegible]

```
    CoreReleaseEventLock ();

    ASSERT (Event->NotifyFunction != NULL);
    Event->NotifyFunction (Event, Event->NotifyContext);
```

- CoreReleaseEventLock ()：[illegible]
- ASSERT (Event->NotifyFunction != NULL)：[illegible]
- Event->NotifyFunction (Event, Event->NotifyContext)：[illegible]

[illegible]

```
    CoreAcquireEventLock ();
  }
```

- CoreAcquireEventLock ()：[illegible]

[illegible]

```
  gEventPending &= ~(UINTN)(1 << Priority);
  CoreReleaseEventLock ();
}
```

- `gEventPending &= ~(UINTN)(1 << Priority)`: [illegible]
- `CoreReleaseEventLock ()`: [illegible]

[illegible]

- [illegible] 31TPL[illegible]
-

[illegible]

Blog

Blog

EDK2 Memory Map

☒☒☒☒☒☒☒☒☒☒☒☒☒

GetMemoryMap

https://blog.csdn.net/xiaopangzi313/article/details/109928878

https://www.lab-z.com/stu5/

☒☒

☒☒☒☒☒

[toc]

☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

Windows ☒☒☒

1. **FAT32☒File Allocation Table 32☒☒**- ☒☒☒☒☒☒☒☒☒☒☒☒☒☒ 2TB☒☒☒☒☒☒☒ 4GB☒☒☒☒☒☒☒☒☒☒☒☒☒

- ☒☒☒☒☒☒☒☒☒☒☒☒☒☒ USB ☒☒☒☒ SD ☒☒

1. **NTFS☒New Technology File System☒☒**- ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

- ☒☒☒☒☒☒☒☒ Windows ☒☒☒☒☒ SSD☒

macOS ☒☒☒

1. **HFS+☒Hierarchical File System Plus☒☒**- ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

- ☒☒☒☒☒☒☒ macOS ☒☒☒☒☒☒☒☒ APFS ☒☒☒

1. **APFS☒Apple File System☒☒**- ☒☒☒☒☒ SSD ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

- ☒☒☒☒☒macOS 10.13 ☒☒☒☒☒☒☒☒☒☒☒☒☒☒ Mac ☒☒☒iOS ☒☒☒

Linux ☒☒☒

1. **ext4☒Fourth Extended File System☒☒**- ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

- ☒☒☒☒☒☒ Linux ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

1. **XFS☒**- ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

- ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

1. **Btrfs☒B-tree File System☒☒**- ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

- ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

☒☒☒☒☒

1. exFAT☒Extended File Allocation Table☒

☒

- ☒☒☒☒ FAT32 ☒ NTFS ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒
- ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ USB ☒☒☒☒☒☒☒☒☒☒☒☒☒ Windows ☒ macOS ☒☒☒☒☒☒☒☒

☒☒☒☒☒☒

1. ZFS[illegible]Zettabyte File System[illegible]

[illegible]

- [illegible] RAID-Z [illegible]
- [illegible]

RAID

[toc]

[illegible]

RAID[illegible]Redundant Array of Independent Disks[illegible] RAID [illegible]

1. [illegible]RAID [illegible]RAID 0[illegible]
2. [illegible] RAID 1[illegible] RAID 5[illegible]RAID 6[illegible]RAID [illegible]RAID 1 [illegible]
3. [illegible]RAID [illegible]RAID 5 [illegible]

[illegible] RAID [illegible]

- **RAID 0**[illegible]
- **RAID 1**[illegible]
- **RAID 5**[illegible] RAID 0[illegible]
- **RAID 6**[illegible] RAID 5[illegible]
- **RAID 10**[illegible]RAID 1+0[illegible] RAID 0 [illegible] RAID 1 [illegible]

0 [illegible]1 [illegible]10 [illegible]

image-20240715131609095

RAID 3

- [illegible]RAID 3 [illegible]
- [illegible]RAID 3 [illegible]
- [illegible]
- [illegible]

RAID 5

- [illegible]RAID 5 [illegible]
- [illegible] RAID 0[illegible]
- [illegible] (frac{(n-1)}{n}) [illegible]n [illegible] RAID 1 [illegible]
- [illegible]
- [illegible] RAID 3 [illegible]

RAID 6

- [illegible]RAID 6 [illegible] RAID 5[illegible]
- [illegible] RAID 6 [illegible] RAID 5 [illegible]

- [illegible] RAID 5 [illegible]
- [illegible] **RAID 5** [illegible] (frac{(n-2)}{n}) [illegible]n [illegible]
- [illegible] RAID 5 [illegible]

[illegible]

- **RAID 3** [illegible]
- **RAID 5** [illegible]
- **RAID 6** [illegible]

SSD

[toc]

[illegible]

SSD [illegible] SOC [illegible]

image-20240715134556835

SSD controller [illegible] FTL [illegible] nand [illegible]

[illegible] PCIE/SATA [illegible] host [illegible] FTL [illegible]

FTL[illegible]flash translation layer[illegible] LBA [illegible] SSD [illegible]

[illegible] flash controller[illegible] ECC[illegible]

[illegible]

[toc]

[illegible]

[illegible]([illegible])[illegible](JBOD)[illegible]

[illegible] [illegible]

[illegible] LUN [illegible]

image-20240715133452696

[illegible]

[illegible] Web [illegible]

[illegible] – [illegible]

[illegible] Ceph [illegible] HDFS [illegible] [illegible] [illegible]

[illegible] CMN SAM [illegible] Ceph[illegible] swift[illegible]

image-20240715133751005

[illegible] [illegible] Ceph [illegible] MON [illegible] OSD [illegible] MDS [illegible]

(1) MON [illegible]MON [illegible]

(2) OSD [illegible] OSD [illegible]

(3) MDS [illegible] CephFS [illegible]Ceph [illegible] RADOS [illegible] Ceph MDS [illegible]

(4) RADOS [illegible]RADOS [illegible] ceph [illegible] Ceph [illegible] RADOS [illegible]RADOS [illegible]

(5) RBD ([illegible])[illegible] RADOS [illegible]

(6) CephFS [illegible]Ceph [illegible] Ceph [illegible] POSIX [illegible]

(7) Librados [illegible]libRADOS [illegible] PHP [illegible] RUBY [illegible] Java [illegible] Python [illegible] C++ [illegible] [illegible] RADOS [illegible]

(8) RADOS GW [illegible]RGW [illegible] Ceph [illegible] RGW [illegible] Amazon S3 [illegible] openstack Swift [illegible] RUSTFUL API

[illegible] RADOS GW [illegible] MON [illegible]([illegible])[illegible] CephFS [illegible]

[illegible]Blog

[illegible]

Hypervisor

[toc]

Introduction

[illegible] Hypervisor [illegible] Type 1 Hypervisor[illegible]

[illegible]

- Type 1 Hypervisor [illegible]
- [illegible] **(OS)**[illegible]
- [illegible] CPU[illegible]
- [illegible]
- [illegible]
- [illegible] Type 1 Hypervisor [illegible]
- [illegible]
- VMware ESXi[illegible]Citrix Hypervisor [illegible] Microsoft Hyper-V [illegible] Type 1 Hypervisor [illegible]

Type 2 Hypervisor

[illegible] 2 [illegible]

[illegible]

- Type 2 hypervisor [illegible]
- [illegible]
- [illegible] CPU[illegible]
- [illegible] (OS) [illegible]
- [illegible]
- [illegible] Type 2 hypervisor [illegible]
- [illegible]
- VMware Workstation Player[illegible]VMware Workstation Pro [illegible] VirtualBox [illegible] Type 2 hypervisor [illegible]

image-20240718133210715

[illegible]

[illegible][illegible] RISC-V [illegible] Type-1 Hypervisor [illegible]

RVirt

RVirt [illegible] MIT PDOS [illegible] 2018 [illegible] Rust [illegible] **RISC-V Type-1 hypervisor**[illegible]RVirt [illegible] RISC-V [illegible] MIT PDOS [illegible]RVirt [illegible] S-mode[illegible] U-mode[illegible] CSR [illegible] hypervisor[illegible]hypervisor [illegible]RVirt [illegible] QEMU virt [illegible] Hifive Unleased [illegible] fedora [illegible]RVirt [illegible] M[illegible]S[illegible]U [illegible] RISC-V [illegible]

Bao-hypervisor

Bao hypervisor [illegible] hypervisor [illegible] Bao hypervisor [illegible]

Bao [illegible]

Bao [illegible] hypervisor[illegible] Bao [illegible]Bao [illegible]

[illegible] Bao [illegible] RISC-V [illegible] Hypervisor Extension [illegible] rocket chip [illegible] CVA6 [illegible]

Xvisor

Xvisor [illegible]Xvisor [illegible] VirtIO [illegible] CPU [illegible] IO [illegible] Xvisor [illegible] Normal vCPUs [illegible]

[illegible] 2019 [illegible] Xvisor [illegible] RISC-V [illegible] RISC-V [illegible] RISC-V Hypervisor Extension 1.0[illegible] Xvisor [illegible] Xvisor [illegible]

seL4

seL4 [illegible] NICTA[illegible] CSIRO data61[illegible]seL4 [illegible] seL4 [illegible]

seL4 [illegible]

- [illegible]seL4 [illegible]
- [illegible]seL4 [illegible]
- [illegible] seL4 [illegible]
- [illegible]seL4 [illegible] GPLv2 [illegible]
- [illegible]seL4 [illegible]
- [illegible] seL4 [illegible] RISC-V [illegible]

KVM

KVM[illegible]Kernel-based Virtual Machine[illegible] Linux [illegible]VM[illegible] VM [illegible]KVM [illegible] Intel VT [illegible] AMD-V[illegible] VM [illegible]

KVM [illegible] Red Hat[illegible]IBM[illegible]Intel [illegible]KVM [illegible] Linux[illegible]Windows [illegible] BSD[illegible]KVM [illegible] virt-manager [illegible] virsh[illegible]

KVM [illegible] Linux [illegible] Linux [illegible] Linux [illegible]

[illegible]

[illegible] RISC-V [illegible] Type-1 Hypervisor [illegible]

MIT6824

6.824 [illegible] MIT [illegible] Golang [illegible]

[illegible]

B [illegible]

[illegible]https://pdos.csail.mit.edu/6.82

Memory consistency and cache coherence

Memory Consistency

[toc]

[illegible]

[illegible]

[illegible] consistency model[illegible] 152 [illegible] 252 [illegible] 152 [illegible] 252 [illegible]“[illegible]”[illegible]

[illegible] consistency model[illegible] FENCE [illegible]

[illegible]

```c
static int x = 0, y = 0;
static int r1, r2;


static void int thread_cpu0(void)
{
        x = 1;    /* 1) */
        r1 = y;   /* 2) */
}


static void int thread_cpu1(void)
{
        y = 1;    /* 3) */
        r2 = x;   /* 4) */
}


static void check_after_assign(void)
{
        printk("r1 = %d, r2 = %d\n", r1, r2);
}
```

[illegible] 4 [illegible] 6 [illegible]

```
1) 2) 3) 4)
1) 3) 2) 4)
1) 3) 4) 2)
3) 4) 1) 2)
```

```
3) 1) 4) 2)
3) 1) 2) 4)
```

[illegible] thread _cpu0 [illegible] thread _cpu1 [illegible] check _after _assign()[illegible] 6 [illegible] 3 [illegible]

```
r1 = 1, r2 = 1
r1 = 0, r2 = 1   // 1) 2) 3) 4)
r1 = 1, r2 = 0   // 3) 4) 1) 2)
```

[illegible] r1 [illegible] r2 [illegible]"[illegible]"[illegible]write buffer[illegible] prefetching [illegible] multiple cache banks [illegible]

[illegible] cache [illegible]write buffer [illegible] cache [illegible] cache [illegible]

image-20240724170156332

[illegible] write buffer[illegible] `1) 3) 2) 4)` [illegible] case[illegible] write buffer [illegible] core [illegible] 0[illegible]

```
r1 = 0, r2 = 0
```

[illegible] `3) 4) 1) 2)` [illegible]

[illegible]

[illegible]

[illegible]**[illegible]**[illegible] TSO [illegible]

[illegible]Memory Consistency Model[illegible]

“[illegible]” —— core 0 [illegible] write buffer [illegible]core1 [illegible]

[illegible]

[illegible] memory consistency and cache coherence[illegible]

cache [illegible] CPU [illegible]

[illegible] CPU [illegible]

[illegible]

[illegible]

[illegible] 4 [illegible] Store-Load[illegible]Store-Store[illegible]Load-Store[illegible]Load-Load[illegible]

[illegible]Sequential Consistency[illegible] SC[illegible] [illegible]Total Store Order[illegible] TSO[illegible]x86-64 [illegible]Part Store Order[illegible] PSO[illegible] [illegible]Relax Memory Order[illegible] RMO[illegible]

[illegible]

[illegible]

1. [illegible]Sequential Consistency, SC[illegible]

[illegible]

[illegible]

- [illegible]
- [illegible]

[illegible]

```
1 -> 3 -> 2 -> 4
3 -> 1 -> 4 -> 2
1 -> 3 -> 4 -> 2 // 创创创创创创创创
3 -> 1 -> 2 -> 4
```

- [illegible]
- [illegible]

[illegible]

1. `1: x = 1` [illegible]CPU 0[illegible]
2. `3: y = 1` [illegible]CPU 1[illegible]
3. `4: r2 = x` [illegible]CPU 1[illegible]
4. `2: r1 = y` [illegible]CPU 0[illegible]

[illegible] `CPU 0` [illegible] `CPU 1` [illegible]

- [illegible] `CPU 0`[illegible] `1 -> 2`
- [illegible] `CPU 1`[illegible] `3 -> 4`

[illegible]

[illegible]

[illegible] `1 -> 3 -> 4 -> 2`[illegible]

- [illegible] 4 [illegible] `x`[illegible] 1 [illegible] `r2 = 1`
- [illegible] 2 [illegible] `y`[illegible] 3 [illegible] `r1 = 1`

2. [illegible]Total Store Order, TSO[illegible]

TSO [illegible]store [illegible] load [illegible] 4 [illegible] store-store[illegible]store-load[illegible]load-load [illegible] load-store[illegible]TSO [illegible] **store-load** [illegible] 3 [illegible]

[illegible]

- [illegible]
- [illegible]

[illegible]

```
1 -> 3 -> 2 -> 4
3 -> 1 -> 4 -> 2
1 -> 2 -> 3 -> 4
3 -> 4 -> 1 -> 2
```

[illegible]

3. [illegible]Partial Store Order, PSO[illegible]

PSO [illegible]

[illegible]Processor Consistency, PC[illegible] [illegible]Relaxed Consistency, RC[illegible]

[illegible]

- [illegible]
- [illegible]

[illegible]

```
1 -> 3 -> 2 -> 4
3 -> 1 -> 4 -> 2
1 -> 2 -> 3 -> 4
3 -> 4 -> 1 -> 2
2 -> 1 -> 4 -> 3
4 -> 3 -> 2 -> 1
```

[illegible]

4. [illegible]Relaxed Memory Order, RMO[illegible]

RMO [illegible]

[illegible]

- [illegible]
- [illegible]

[illegible]

```
1 -> 3 -> 2 -> 4
3 -> 1 -> 4 -> 2
```

```
1 -> 2 -> 3 -> 4
3 -> 4 -> 1 -> 2
2 -> 1 -> 4 -> 3
4 -> 3 -> 2 -> 1
2 -> 4 -> 1 -> 3
4 -> 2 -> 3 -> 1
```

[illegible]

[illegible]

[illegible]	SC	TSO	PSO	RMO
Store-Store	[illegible]	[illegible]	Yes	Yes
Store-Load	[illegible]	Yes	Yes	Yes
Load-Load	[illegible]	[illegible]	[illegible]	Yes
Load-Store	[illegible]	[illegible]	[illegible]	Yes

[illegible]

[illegible] `r1 = 0, r2 = 0` [illegible]Linux [illegible] `smp_mb()`[illegible]`smp_mb()`[illegible]

[illegible]

[illegible]

- [A Primer on Memory Consistency and Cache Coherence (2nd Edition)]
- https://blog.csdn.net/qq _29328443/article/details/107616795
- https://blog.csdn.net/anyegongjuezjd/article/details/125954805
- https://zhuanlan.zhihu.com/p/141655129?from _voters _page=true
- [illegible]

□□

ARM □□□□□□

1. □□□□□□□□□□BMC □□□□□□□ C2000 CHIP □ POR _N□pervsoc □□□□□□□□□□
2. □ otprom valid □□□□□□□ SCP SRAM □ BISR □□□□PVO(SCP)□□□ ROM □otprom □□□□□□□□□□
3. □ SCP □□□□□□ SRAM□
4. SCP ROM FW □ GPIO □□□□□□ CHIP ID,□□ CHIP ID □□□□□□□□ CHIP;
5. □□□□□□□□□□□ I2C/FSI□□□□□□□□□□□□□□□
6. □ CHIP □ SCP □□□□□□ QSPI;
7. □ CHIP □ SCP □□□□□ CHIP □□□□□□□□□□□□□ CHIP □□□
8. □ CHIP □ SCP □□□□□ CHIP SCP □□□□□□□
9. □ CHIP □ SCP □ QSPI □□□□□ SCP RAM FW □□ CHIP □ SRAM;
10. □ CHIP □ SCP □□□□ 8 □□□□□□□□□□ SCP RAM FW □□□□□□□□□□ SCP,□ SCP □□□□□□□ SRAM:
11. □ CHIP □ SCP □□□□□ SRAM,□□ SCP RAM FW:
12. □ CHIP □ SCP □□ QSPI □□ C2C □□□□□□□□□□□□□□□ SCP□
13. □ CHIP □ SCP □□ CHIP □ SCP □□□□□ C2C □□□□□□ C2C;
14. □ SCP □□ CHIP ID □□□ CMN□OCM(SLC)□ SAM;
15. □ CHIP □ SCP □□□□□ MCP RAM FW,□□ MCP(□□□)□+
16. □ SCP □□□□□ CHIP □□□□ L2□NCU□SCOM/FIR;+
17. □ CHIP □ SCP □□□□□ Hostboot base □□□□ CHIP □ OCM(SLC);+
18. □ CHIP □ SCP □□□ CHIP □□□□□□□□□□□□□□□□
19. □□□□□□□□□□□ srest□0x100□□□□□□

□ core-hostboot

1. □□ kernel other thread spinlock,□□□□□□□□ thread □□□□
2. □□ hostboot kernel □□□□□□□□□ C/C++□□ main □□□+
3. □□ hostboot kernel □□□□□□□□□□□
4. □□□ MMU(SLB/TLB),□□ DIMM □□□□□□□□ OCM(SLC,□□□□□□ 28 □* 2MB=56MB□□□□□ 3/4□□
5. □□□□□□□□ TB □□□ CPU □□□
6. □ Scratch □□□□□□ FSP/BMC;
7. □□ CPU □□ init main □□□

8. ⊠⊠ kernel other thread spinlock ⊠ 1⊠⊠⊠⊠⊠ thread ⊠⊠⊠⊠
9. hostboot ⊠⊠ task ⊠⊠⊠⊠ task ⊠⊠⊠ user ⊠⊠⊠⊠ kernel ⊠⊠⊠⊠ hypersior ⊠⊠⊠
10. init main ⊠⊠⊠⊠⊠ VFS;
11. VFS ⊠⊠ base modules⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

⊠ init ⊠⊠⊠

1. ⊠⊠ initservice ⊠⊠⊠
2. initservice ⊠ g _taskinfolist ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠
3. secureboot _base/ipmibase/pnor/vfs/extinitsvc ⊠⊠⊠⊠⊠⊠⊠⊠
4. ⊠⊠⊠⊠⊠⊠ hostboot extended image ⊠⊠⊠⊠⊠⊠⊠
5. extinitsvc ⊠⊠⊠ g _exttaskinfolist ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠
6. ⊠⊠⊠⊠⊠⊠ istepdisp⊠⊠⊠⊠⊠ hostboot ⊠⊠⊠⊠⊠⊠⊠
7. ⊠ istep ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ hostboot ⊠⊠⊠⊠
8. istep 21.03 ⊠⊠⊠⊠⊠⊠⊠ payload(skiboot,OPAL);+
9. ⊠⊠ payload(skiboot,OPAL)⊠

⊠ core-skiboot

1. ⊠⊠⊠⊠⊠⊠⊠⊠C2000 ⊠⊠ skiboot ⊠⊠⊠⊠⊠⊠
2. ⊠ thread ⊠⊠ skiboot ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ relocate ⊠⊠⊠⊠⊠ thread ⊠ ⊠⊠⊠
3. ⊠ thread ⊠⊠ boot flag=1,⊠⊠⊠⊠ thread ⊠⊠⊠⊠
4. ⊠ thread ⊠⊠ main cpu entry,⊠⊠ thread ⊠⊠⊠ secondary cpu entry;
5. ⊠⊠⊠ uart(console);
6. ⊠⊠⊠ skiboot ⊠⊠⊠⊠⊠⊠⊠/⊠⊠⊠⊠⊠
7. ⊠⊠ skiboot ⊠⊠⊠⊠⊠⊠⊠⊠
8. ⊠⊠⊠ opal call table;
9. ⊠⊠ hostboot ⊠⊠⊠⊠ device tree ⊠ HDAT;
10. ⊠⊠ DT ⊠⊠⊠ SCOM/eSPI(LPC):
11. ⊠⊠ DT ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠C2000 ⊠ DIMM ⊠⊠⊠⊠⊠⊠⊠⊠
12. ⊠⊠ DT ⊠⊠⊠ homer⊠C2000 ⊠⊠⊠⊠⊠⊠⊠⊠

13.⊠⊠ DT ⊠⊠⊠⊠⊠ CPU(thread);

14.⊠ opal ⊠⊠⊠⊠ DT;

15.⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

16.⊠⊠⊠⊠⊠⊠⊠⊠⊠ BMC ⊠⊠⊠ ipmi/BT/UART;

17.⊠⊠⊠⊠⊠⊠⊠GIC⊠

18.⊠⊠⊠⊠ CPU⊠thread⊠⊠⊠⊠⊠⊠

19.⊠⊠ sreset⊠⊠⊠⊠ CPU ⊠⊠⊠⊠⊠⊠

20.⊠⊠⊠ TOD⊠⊠⊠ TOD ⊠⊠⊠

21.⊠⊠⊠ sensor ⊠⊠⊠ OPAL ⊠⊠⊠

22.⊠⊠⊠⊠⊠⊠⊠⊠

23.NV ⊠⊠⊠⊠

24.⊠⊠⊠⊠jiayu⊠⊠⊠⊠

25.⊠⊠⊠⊠TPM⊠⊠⊠⊠

26.⊠⊠⊠ opal console(UART)⊠⊠⊠

1. PCIE HB ⊠⊠⊠⊠ PCIE ⊠⊠⊠ (RC ⊠⊠⊠⊠CXL ⊠ PCIe ⊠⊠⊠device tree ⊠⊠⊠⊠⊠)

28.PCIE SLOT ⊠⊠⊠⊠

29.⊠⊠⊠⊠⊠⊠

30.⊠⊠⊠⊠⊠⊠⊠

31.⊠⊠⊠⊠⊠⊠ zImage(boot loader ⊠⊠⊠ petitboot);

32.⊠⊠ boot loader ⊠⊠⊠

image-20240704111929075

image-20240704111945789

CMN OCM(on chip memory) ⊠ L3 cache⊠⊠⊠⊠ DDR ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ Intel ⊠ cache as ram ⊠⊠⊠

⊠ OCM ⊠⊠ memory ⊠⊠⊠⊠⊠CMN OCM(on chip memory) ⊠⊠⊠⊠ 28*2=56M ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ 3/4⊠⊠⊠⊠ ⊠⊠ 32M⊠

⊠⊠⊠⊠

SCP ram ⊠⊠⊠ DDR training⊠⊠⊠⊠⊠ OCM ⊠⊠⊠ flash ⊠⊠⊠⊠⊠⊠ DDR ⊠⊠⊠⊠

⊠⊠⊠⊠ OCM ⊠⊠⊠⊠⊠⊠⊠⊠ SNF ⊠⊠⊠⊠

⊠⊠⊠⊠

SCP ⊠⊠⊠ OCM ⊠⊠⊠SCP RAM ⊠ hostboot ⊠⊠ OCM ⊠⊠⊠⊠skiboot ⊠⊠⊠⊠ OCM ⊠⊠⊠⊠ skiboot ⊠⊠⊠ ddr training⊠⊠⊠⊠⊠⊠ DDR⊠⊠⊠ OCM⊠⊠⊠ OCM⊠⊠⊠ cache ⊠⊠⊠

hostboot ⊠⊠ skiboot ⊠⊠ 0x800 0000 ⊠⊠

32M ⊠⊠⊠⊠⊠ 0x200 0000, ⊠ skiboot ⊠⊠⊠⊠⊠⊠ 0x3000 0000 ⊠⊠⊠⊠⊠⊠ OCM⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ 0x0⊠⊠⊠ ⊠⊠⊠ 0x0 ⊠⊠⊠

RISC-V [illegible]

[illegible]

Firmware [illegible]

image-20240711102511545

ARM64 vs RISC-V

image-20240711103010418

uboot [illegible] EDK2[illegible] PEI [illegible] DXE [illegible]

image-20240711103301123

[illegible]

ZSBL: SCP ROM

FSBL: SCP RAM

image-20240711105826248

Cache as RAM & On chip memory

☒☒

☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

x86 ☒☒☒ Cache as RAM, CAR;

ARM ☒☒☒☒ On Chip Memory☒OCM☒

☒☒☒☒☒☒☒☒☒☒ DDR ☒☒☒☒☒☒☒☒☒☒☒ cache ☒☒ RAM☒☒☒☒ C ☒☒☒☒☒

CAR

EDK2 System Table

image-20240729101921741

⊠ `MdeModulePkg\Core\Dxe\DxeMain\DxeMain.c` ⊠⊠⊠⊠

```
  //
  // Allocate the EFI System Table and EFI Runtime Service Table from
EfiRuntimeServicesData
  // Use the templates to initialize the contents of the EFI System Table and EFI
Runtime Services Table
  //
  gDxeCoreST = AllocateRuntimeCopyPool (sizeof (EFI_SYSTEM_TABLE),
&mEfiSystemTableTemplate);
  ASSERT (gDxeCoreST != NULL);

  gDxeCoreRT = AllocateRuntimeCopyPool (sizeof (EFI_RUNTIME_SERVICES),
&mEfiRuntimeServicesTableTemplate);
  ASSERT (gDxeCoreRT != NULL);

  gDxeCoreST->RuntimeServices = gDxeCoreRT;
```

⊠⊠⊠

```
///
/// EFI System Table
///
typedef struct {
  ///
  /// The table header for the EFI System Table.
  ///
  EFI_TABLE_HEADER                  Hdr;
  ///
  /// A pointer to a null terminated string that identifies the vendor
  /// that produces the system firmware for the platform.
  ///
  CHAR16                            *FirmwareVendor;
  ///
  /// A firmware vendor specific value that identifies the revision
  /// of the system firmware for the platform.
  ///
  UINT32                            FirmwareRevision;
  ///
  /// The handle for the active console input device. This handle must support
  /// EFI_SIMPLE_TEXT_INPUT_PROTOCOL and EFI_SIMPLE_TEXT_INPUT_EX_PROTOCOL.
  ///
  EFI_HANDLE                        ConsoleInHandle;
  ///
  /// A pointer to the EFI_SIMPLE_TEXT_INPUT_PROTOCOL interface that is
  /// associated with ConsoleInHandle.
  ///
  EFI_SIMPLE_TEXT_INPUT_PROTOCOL    *ConIn;
  ///
  /// The handle for the active console output device.
  ///
  EFI_HANDLE                        ConsoleOutHandle;
  ///
  /// A pointer to the EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL interface
  /// that is associated with ConsoleOutHandle.
  ///
  EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL   *ConOut;
  ///
  /// The handle for the active standard error console device.
  /// This handle must support the EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL.
  ///
  EFI_HANDLE                        StandardErrorHandle;
  ///
  /// A pointer to the EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL interface
  /// that is associated with StandardErrorHandle.
  ///
  EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL   *StdErr;
  ///
  /// A pointer to the EFI Runtime Services Table.
  ///
  EFI_RUNTIME_SERVICES              *RuntimeServices;
  ///
  /// A pointer to the EFI Boot Services Table.
  ///
  EFI_BOOT_SERVICES                 *BootServices;
  ///
  /// The number of system configuration tables in the buffer ConfigurationTable.
  ///
  UINTN                             NumberOfTableEntries;
  ///
  /// A pointer to the system configuration tables.
  /// The number of entries in the table is NumberOfTableEntries.
  ///
  EFI_CONFIGURATION_TABLE           *ConfigurationTable;
} EFI_SYSTEM_TABLE;
```

EDK2 Event

[toc]

[illegible]

[illegible]

[illegible]UEFI [illegible]

[illegible] [illegible]/[illegible] [illegible] intel [illegible] 8253/8254 [illegible] PC [illegible] IC[illegible] MSC8051,[illegible]

[illegible] Timer [illegible] cpu [illegible]

[illegible] PC/AT [illegible] UEFI [illegible] 8254 [illegible] Timer[illegible] mode 3[illegible] 1.1931816MHz[illegible] tick [illegible]mTimerPeriod[illegible] 1ms[illegible]10000 [illegible] tick [illegible]=10000*100ns[illegible] CPU [illegible] IRQ0[illegible] 0 [illegible] OUT pin [illegible] 8259 [illegible] IRQ0 [illegible] CPU [illegible] Timer [illegible]

[illegible] cpu archprotocol [illegible] Legacy8259Protocol[illegible],[illegible]cpu [illegible] IDT [illegible] Timer [illegible] TimerInterruptHandler[illegible]

Timer [illegible]

[illegible] Timer [illegible]

1.[illegible]

2.[illegible]

3.[illegible]

4.[illegible] Timer [illegible] CoreTimerTick(mTimerPeriod)/CoreTimerTick(100ms)[illegible] TimerArchprotocol [illegible] install [illegible] EVT _NOTIFY _SIGNAL [illegible] Event [illegible]

5.[illegible] Sytem time [illegible] mEfiSystemTime =mEfiSystemTime+ 1Tick(100ms)

6. [illegible] timer event list [illegible] event[illegible] BS->SignalEvent [illegible] signal [illegible]

BS->SignalEvent—>CoreNotifyEvent–>CoreRestoreTpl–>CoreDispatchEventNotifies–>A.[illegible] EVT _NOTIFY _SIGNAL [illegible] count[illegible]Event->SignalCount = 0[illegible] Event->NotifyFunction (Event, Event->NotifyContext)[illegible] Tpl [illegible] gEventPending mask bit[illegible] event[illegible]

7.[illegible]

8.[illegible]

[illegible]

1. gEventQueue— [illegible]A list of event’s to notify for each priority level[illegible]
2. mEfiTimerList—-[illegible] timer [illegible] event [illegible]

3. gEventSignalQueue —⊠⊠⊠⊠ A list of events to signal based on EventGroup type

4. ⊠⊠ EVT _NOTIFY _WAIT ⊠⊠⊠ event⊠⊠⊠⊠⊠ BS-> **CoreCheckEvent**->CoreNotifyEvent–>CoreRestoreTpl–>CoreDispatchEventNotifies ⊠⊠⊠⊠⊠⊠⊠⊠⊠ gEventQueue ⊠⊠ IEvent ⊠⊠⊠⊠

gEventPending ⊠⊠⊠⊠⊠⊠⊠ gEventQueue ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ 1⊠⊠⊠⊠ 0⊠

gEventPending ⊠⊠⊠ 32 ⊠⊠⊠⊠⊠⊠⊠⊠ 1 ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

⊠⊠⊠ x86 ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠IRQ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

- IRQ0: ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠
- IRQ1: ⊠⊠
- IRQ2: ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠
- IRQ3: ⊠⊠⊠⊠ 2⊠COM2⊠
- IRQ4: ⊠⊠⊠⊠ 1⊠COM1⊠
- IRQ5: ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠
- IRQ6: ⊠⊠⊠⊠⊠
- IRQ7: ⊠⊠⊠⊠ 1⊠LPT1⊠
- ⊠⊠ IRQ ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

event ⊠⊠⊠

```
//
// Task priority level
//
#define TPL_APPLICATION       4
#define TPL_CALLBACK          8
#define TPL_NOTIFY            16
#define TPL_HIGH_LEVEL        31
```

31 ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

⊠⊠⊠⊠⊠

image-20240716144753316

⊠ DXE main ⊠⊠⊠⊠

```
  //
  // Initialize the Event Services
  //
  Status = CoreInitializeEventServices ();

...

  //
  // Register for the GUIDs of the Architectural Protocols, so the rest of the
  // EFI Boot Services and EFI Runtime Services tables can be filled in.
  // Also register for the GUIDs of optional protocols.
  //
  CoreNotifyOnProtocolInstallation ();
```

[illegible] event [illegible]

[illegible]

```
/**
  Called by the platform code to process a tick.

  @param  Duration               The number of 100ns elapsed since the last call
                                 to TimerTick

**/
VOID
EFIAPI
CoreTimerTick (
  IN UINT64   Duration
  )
{
  IEVENT          *Event;

  //
  // Check runtiem flag in case there are ticks while exiting boot services
  //
  CoreAcquireLock (&mEfiSystemTimeLock);

  //
  // Update the system time
  //
  mEfiSystemTime += Duration;

  //
  // If the head of the list is expired, fire the timer event
  // to process it
  //
  if (!IsListEmpty (&mEfiTimerList)) {
    Event = CR (mEfiTimerList.ForwardLink, IEVENT, Timer.Link, EVENT_SIGNATURE);

    if (Event->Timer.TriggerTime <= mEfiSystemTime) {
      CoreSignalEvent (mEfiCheckTimerEvent);
    }
  }

  CoreReleaseLock (&mEfiSystemTimeLock);
}
```

[illegible] timer [illegible] event[illegible]

```
/**
  Initializes "event" support.

  @retval EFI_SUCCESS            Always return success

**/
EFI_STATUS
CoreInitializeEventServices (
  VOID
  )
{
  UINTN        Index;

  for (Index=0; Index <= TPL_HIGH_LEVEL; Index++) {
    InitializeListHead (&gEventQueue[Index]);
  }

  CoreInitializeTimer ();

  CoreCreateEventEx (
    EVT_NOTIFY_SIGNAL,
    TPL_NOTIFY,
    EfiEventEmptyFunction,
    NULL,
    &gIdleLoopEventGuid,
    &gIdleLoopEvent
    );

  return EFI_SUCCESS;
}
```

- [illegible] gEventQueue— [illegible]A list of event's to notify for each priority level
- CoreInitializeTimer [illegible]

```
/**
  Initializes timer support.

**/
VOID
CoreInitializeTimer (
  VOID
  )
{
  EFI_STATUS  Status;

  Status = CoreCreateEventInternal (
             EVT_NOTIFY_SIGNAL,
             TPL_HIGH_LEVEL - 1,
             CoreCheckTimers,
             NULL,
             NULL,
             &mEfiCheckTimerEvent
             );
  ASSERT_EFI_ERROR (Status);
}
```

☒☒☒☒☒☒☒☒

```
#include <Uefi.h>
#include <Library/UefiBootServicesTableLib.h>
#include <Library/UefiLib.h>

EFI_EVENT TimerEvent;
EFI_EVENT ExitBootServicesEvent;
BOOLEAN TimerTriggered = FALSE;

VOID EFIAPI TimerEventHandler(
    IN EFI_EVENT Event,
    IN VOID *Context
) {
    TimerTriggered = TRUE;
    Print(L"Timer event triggered!\n");
}

EFI_STATUS
EFIAPI
UefiMain(
    IN EFI_HANDLE ImageHandle,
    IN EFI_SYSTEM_TABLE *SystemTable
) {
    EFI_STATUS Status;

    // □□□□□□□□□□□□□□10000000 100ns (1□)□□□
    Status = gBS->CreateEvent(
        EVT_TIMER | EVT_NOTIFY_SIGNAL,
        TPL_CALLBACK,
        TimerEventHandler,
        NULL,
        &TimerEvent
    );
    if (EFI_ERROR(Status)) {
        Print(L"Failed to create event: %r\n", Status);
        return Status;
    }
    // □□□□□□1 □□□□
    Status = gBS->SetTimer(
        TimerEvent,
        TimerRelative,
        10000000 // 100ns □□□10000000 = 1 □
    );
    if (EFI_ERROR(Status)) {
        Print(L"Failed to set timer: %r\n", Status);
        gBS->CloseEvent(TimerEvent);
        return Status;
    }

    // □□□□□□
    while (!TimerTriggered) {
        gBS->Stall(100000); // □□ 100ms
    }

    // □□
    gBS->CloseEvent(TimerEvent);

    return EFI_SUCCESS;
}
```

□□□□□□□□□□□□□□1 □□□□□

□□□□□ gBS->CreateEvent □□□□□□□□□□□□□□□□□□□□□□

```
typedef struct {
    UINT32 Type;
    EFI_TPL NotifyTpl;
    EFI_EVENT_NOTIFY NotifyFunction;
    VOID *NotifyContext;
    BOOLEAN Triggered;
} EFI_EVENT_INTERNAL;

EFI_STATUS
EFIAPI
CreateEvent (
    IN UINT32 Type,
    IN EFI_TPL NotifyTpl,
    IN EFI_EVENT_NOTIFY NotifyFunction,
    IN VOID *NotifyContext,
    OUT EFI_EVENT *Event
) {
    EFI_STATUS Status;
    EFI_EVENT_INTERNAL *NewEvent;

    // □□□□
    if (Event == NULL || (Type & EVT_NOTIFY_SIGNAL && NotifyFunction == NULL)) {
        return EFI_INVALID_PARAMETER;
    }

    // [illegible]
    NewEvent = AllocateZeroPool(sizeof(EFI_EVENT_INTERNAL));
    if (NewEvent == NULL) {
        return EFI_OUT_OF_RESOURCES;
    }

    // [illegible]
    NewEvent->Type = Type;
    NewEvent->NotifyTpl = NotifyTpl;
    NewEvent->NotifyFunction = NotifyFunction;
    NewEvent->NotifyContext = NotifyContext;
    NewEvent->Triggered = FALSE;

    // [illegible]
    Status = RegisterEvent(NewEvent);
    if (EFI_ERROR(Status)) {
        FreePool(NewEvent);
        return Status;
    }

    // [illegible]
    *Event = (EFI_EVENT)NewEvent;

    return EFI_SUCCESS;
}
```

[illegible] event [illegible]

[illegible] event [illegible]

```
  if ((Type & EVT_NOTIFY_SIGNAL) != 0x00000000) {
    //
    // The Event's NotifyFunction must be queued whenever the event is signaled
    //
    InsertHeadList (&gEventSignalQueue, &IEvent->SignalLink);
  }
```

[illegible] 32 [illegible] bitmap [illegible] 1[illegible]

[illegible]

SignalEvent [illegible] event [illegible] 32 [illegible]

```
/**
  Signals the event.  Queues the event to be notified if needed.
  @param  UserEvent              The event to signal .

  @retval EFI_INVALID_PARAMETER  Parameters are not valid.
  @retval EFI_SUCCESS            The event was signaled.

**/
EFI_STATUS
EFIAPI
CoreSignalEvent (
  IN EFI_EVENT    UserEvent
  )
/**
  Queues the event's notification function to fire.

  @param  Event                  The Event to notify

**/
VOID
CoreNotifyEvent (
  IN  IEVENT      *Event
  )
{

  //
  // Event database must be locked
  //
  ASSERT_LOCKED (&gEventQueueLock);

  //
  // If the event is queued somewhere, remove it
  //

  if (Event->NotifyLink.ForwardLink != NULL) {
    RemoveEntryList (&Event->NotifyLink);
    Event->NotifyLink.ForwardLink = NULL;
  }

  //
  // Queue the event to the pending notification list
  //

  InsertTailList (&gEventQueue[Event->NotifyTpl], &Event->NotifyLink);
  gEventPending |= (UINTN)(1 << Event->NotifyTpl);
}
```

[illegible]

Duang[illegible]CoreTimerTick [illegible] TPL[illegible] restore TPL[illegible]

[illegible] timer [illegible] event[illegible]mEfiTimerList[illegible]

[illegible] TPL [illegible]

```
  while (gEventPending != 0)
{
    PendingTpl = (UINTN) HighBitSet64 (gEventPending);
    if (PendingTpl <= NewTpl) {
      break;
    }
    gEfiCurrentTpl = PendingTpl;
    if (gEfiCurrentTpl < TPL_HIGH_LEVEL) {
      CoreSetInterruptState (TRUE);
    }
    CoreDispatchEventNotifies (gEfiCurrentTpl);
    //创创 gEventQueue [gEfiCurrentTpl]创创创创创 event 创 Notification 创创
  }
```

[illegible]gEventPending [illegible] 32 [illegible] 1 [illegible]

[illegible]

1. [illegible]
2. [illegible] TPL [illegible] TPL [illegible] 16[illegible] 17[illegible] 16 [illegible] TPL [illegible] 17 [illegible]
3. [illegible] pending [illegible] pending [illegible] TPL [illegible] 16[illegible] 15[illegible] 16 [illegible]

[illegible]——[illegible]

[illegible] waitforevent [illegible] 4[illegible] 10ms [illegible] 31[illegible] RestoreTpl [illegible] 4, [illegible] 31 [illegible] 4 [illegible] 30 [illegible] 5 [illegible] 4 [illegible]

CoreTimerTick [illegible]

```
/**
  Lowers the task priority to the previous value.   If the new
  priority unmasks events at a higher priority, they are dispatched.

  @param  NewTpl  New, lower, task priority

**/
VOID
EFIAPI
CoreRestoreTpl (
  IN EFI_TPL NewTpl
  )
```

☒☒☒☒

```
{
  EFI_TPL      OldTpl;
  EFI_TPL      PendingTpl;
```

- OldTpl: ☒☒☒☒☒☒☒☒☒☒
- PendingTpl: ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

☒☒☒☒☒☒

```
  OldTpl = gEfiCurrentTpl;
  if (NewTpl > OldTpl) {
    DEBUG ((EFI_D_ERROR, "FATAL ERROR - RestoreTpl with NewTpl(0x%x) > OldTpl(0x%x)\n",
NewTpl, OldTpl));
    ASSERT (FALSE);
  }
  ASSERT (VALID_TPL (NewTpl));
```

- OldTpl = gEfiCurrentTpl: ☒☒☒☒☒☒☒☒☒☒
- ☒☒ NewTpl ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ NewTpl ☒☒ OldTpl☒☒☒☒☒☒☒☒☒☒☒☒☒☒

☒☒☒☒☒☒

```
  if (OldTpl >= TPL_HIGH_LEVEL  &&  NewTpl < TPL_HIGH_LEVEL) {
    gEfiCurrentTpl = TPL_HIGH_LEVEL;
  }
```

- ☒☒☒☒☒☒☒☒☒☒☒☒☒TPL_HIGH_LEVEL☒☒☒☒☒☒☒☒☒☒☒☒☒ TPL_HIGH_LEVEL☒☒☒☒☒☒☒☒☒☒☒ TPL_HIGH_LEVEL☒☒☒☒ ☒☒☒☒☒☒

☒☒☒☒☒☒☒

```
  while (gEventPending != 0) {
    PendingTpl = (UINTN) HighBitSet64 (gEventPending);
    if (PendingTpl <= NewTpl) {
      break;
    }

    gEfiCurrentTpl = PendingTpl;
    if (gEfiCurrentTpl < TPL_HIGH_LEVEL) {
      CoreSetInterruptState (TRUE);
    }
    CoreDispatchEventNotifies (gEfiCurrentTpl);
  }
```

- while (gEventPending != 0): [illegible]
- PendingTpl = (UINTN) HighBitSet64 (gEventPending): [illegible]
- [illegible] PendingTpl [illegible] NewTpl[illegible]
- [illegible] PendingTpl[illegible] TPL_HIGH_LEVEL [illegible]
- [illegible] CoreDispatchEventNotifies [illegible]

[illegible]

```
  gEfiCurrentTpl = NewTpl;

  if (gEfiCurrentTpl < TPL_HIGH_LEVEL) {
    CoreSetInterruptState (TRUE);
  }

}
```

- [illegible] NewTpl[illegible]
- [illegible] TPL_HIGH_LEVEL[illegible]

CoreDispatchEventNotifies [illegible]

```
/**
  Dispatches all pending events.

  @param  Priority               The task priority level of event notifications
                                 to dispatch

**/
VOID
CoreDispatchEventNotifies (
  IN EFI_TPL      Priority
  )
```

[illegible]

```
{
  IEVENT          *Event;
  LIST_ENTRY      *Head;
```

- Event: [illegible]
- Head: [illegible]

[illegible]

```
CoreAcquireEventLock ();
ASSERT (gEventQueueLock.OwnerTpl == Priority);
Head = &gEventQueue[Priority];
```

- CoreAcquireEventLock (): [illegible]
- ASSERT (gEventQueueLock.OwnerTpl == Priority): [illegible]
- Head = &gEventQueue[Priority]: [illegible]

[illegible]

```
while (!IsListEmpty (Head)) {

  Event = CR (Head->ForwardLink, IEVENT, NotifyLink, EVENT_SIGNATURE);
  RemoveEntryList (&Event->NotifyLink);

  Event->NotifyLink.ForwardLink = NULL;
```

- while (!IsListEmpty (Head)): [illegible]
- Event = CR (Head->ForwardLink, IEVENT, NotifyLink, EVENT_SIGNATURE): [illegible]
- RemoveEntryList (&Event->NotifyLink): [illegible]
- Event->NotifyLink.ForwardLink = NULL: [illegible]

[illegible] SIGNAL [illegible]

```
  if ((Event->Type & EVT_NOTIFY_SIGNAL) != 0) {
    Event->SignalCount = 0;
  }
```

- [illegible] EVT_NOTIFY_SIGNAL[illegible] SignalCount[illegible]

[illegible]

```
  CoreReleaseEventLock ();

  ASSERT (Event->NotifyFunction != NULL);
  Event->NotifyFunction (Event, Event->NotifyContext);
```

- CoreReleaseEventLock (): [illegible]
- ASSERT (Event->NotifyFunction != NULL): [illegible]
- Event->NotifyFunction (Event, Event->NotifyContext): [illegible]

[illegible]

```
  CoreAcquireEventLock ();
}
```

- CoreAcquireEventLock (): [illegible]

[illegible]

```
  gEventPending &= ~(UINTN)(1 << Priority);
  CoreReleaseEventLock ();
}
```

- `gEventPending &= ~(UINTN)(1 << Priority)`: [illegible]
- `CoreReleaseEventLock ()`: [illegible]

[illegible]

- [illegible] 31TPL[illegible]
-

[illegible]

Blog

Blog

EDK2 Memory Map

[illegible]

GetMemoryMap

https://blog.csdn.net/xiaopangzi313/article/details/109928878

https://www.lab-z.com/stu5/

OS

FreeRTOS

[TOC]

[illegible]

[illegible]

[illegible]

1. [illegible]Determinism[illegible]

RTOS [illegible]RTOS [illegible]

2. [illegible]Low Latency[illegible]

RTOS [illegible]

3. [illegible]Priority Scheduling[illegible]

RTOS [illegible] CPU [illegible]Priority Preemptive Scheduling[illegible]Round-Robin Scheduling[illegible]

4. [illegible]Interrupt Handling[illegible]

RTOS [illegible]ISR[illegible]

5. [illegible]Time Management[illegible]

RTOS [illegible]

6. [illegible]Resource Management[illegible]

RTOS [illegible]

Linux [illegible]

[illegible]

1. [illegible]

- [illegible] Linux [illegible]
- [illegible] Linux [illegible]

2. [illegible]

- [illegible]Linux [illegible] CPU [illegible] Linux [illegible]SCHED _FIFO [illegible] SCHED _RR[illegible]
- [illegible]Linux [illegible]

3. [illegible]

- [illegible]Linux [illegible]
- [illegible]**Swapping**[illegible]Linux [illegible]

4. [illegible]

- [illegible]Linux [illegible]
- [illegible]

5. [illegible]

- [illegible]Linux [illegible] CPU[illegible]
- [illegible]

6. [illegible]

- **[illegible] vs [illegible]**[illegible]Linux [illegible]Linux [illegible]

☒☒☒☒☒☒☒

ACPI

ACPI Introduction and Overview

[toc]

History of ACPI

ACPI [illegible] 1990 [illegible]Phoenix [illegible] ACPI [illegible] BIOS([illegible])[illegible] BIOS [illegible]BIOS [illegible]/[illegible](I/O)[illegible]([illegible]plug and play)[illegible] BIOS [illegible]

[illegible] BIOS [illegible]

[illegible] ACPI [illegible]

What is ACPI?

ACPI [illegible]([illegible])[illegible]

[illegible] ACPI [illegible]ACPI [illegible](BIOS)[illegible]"[illegible] 1"[illegible]

image-20240729095600016

[illegible]ACPI [illegible](data table)[illegible](definition block)[illegible]"[illegible] 2"[illegible]

image-20240729095720716

[illegible] ASL(ACPI Source Language) [illegible]ASL [illegible] ACPI [illegible]ASL [illegible] ASL [illegible] AML(ACPI [illegible]) [illegible]AML [illegible] ACPI AML [illegible]"[illegible] 3"[illegible]

image-20240729095803629

AML [illegible]AML [illegible]/[illegible]I/O[illegible]PCI [illegible]"[illegible]"[illegible] AML [illegible]

[illegible]"ACPI [illegible]"[illegible] ACPI [illegible] ACPI [illegible]([illegible] PCI [illegible] USB [illegible])[illegible] ACPI [illegible] ACPI [illegible] ACPI [illegible]

[illegible] _HID [illegible]([illegible])[illegible] ACPI [illegible] _ADR [illegible]([illegible])[illegible] ACPI [illegible] ACPI [illegible] _ADR [illegible] ACPI [illegible]ACPI [illegible]

[illegible]PCI [illegible]PCI [illegible] ACPI [illegible] ACPI [illegible] PCI [illegible]

ACPI [illegible] ACPI [illegible]ACPI [illegible]

ACPI initialization

[illegible] ACPI [illegible]"[illegible] 4″[illegible]:

image-20240729100047387

[illegible] ACPI [illegible](XSDT)[illegible] ACPI [illegible] ACPI [illegible]XSDT [illegible] ACPI [illegible](FADT)[illegible]FADT [illegible] ACPI [illegible](DSDT)[illegible]

ACPI [illegible] DSDT [illegible] ACPI [illegible]XSDT [illegible](SSDT)[illegible]ACPI [illegible]

[illegible] ACPI [illegible] _HID [illegible]

image-20240729100840760

ACPI [illegible] EDK2 [illegible]

system table -> system configuration table -> ACPI table

system table [illegible] OS[illegible] OS [illegible]

[illegible] ACPI table [illegible] system configuration table [illegible] entry[illegible]entry [illegible] GUID [illegible]

```
///
/// Contains a set of GUID/pointer pairs comprised of the ConfigurationTable field in
the
/// EFI System Table.
///
typedef struct {
  ///
  /// The 128-bit GUID value that uniquely identifies the system configuration table.
  ///
  EFI_GUID                          VendorGuid;
  ///
  /// A pointer to the table associated with VendorGuid.
  ///
  VOID                              *VendorTable;
} EFI_CONFIGURATION_TABLE;
```

ACPI [illegible] GUID [illegible] `MdePkg\Include\Guid\Acpi.h`

```
#define ACPI_TABLE_GUID \
  { \
    0xeb9d2d30, 0x2d88, 0x11d3, {0x9a, 0x16, 0x0, 0x90, 0x27, 0x3f, 0xc1, 0x4d } \
  }
```

```
#define EFI_ACPI_TABLE_GUID \
  { \
    0x8868e871, 0xe4f1, 0x11d3, {0xbc, 0x22, 0x0, 0x80, 0xc7, 0x3c, 0x88, 0x81 } \
  }

#define ACPI_10_TABLE_GUID     ACPI_TABLE_GUID

//
// ACPI 2.0 or newer tables should use EFI_ACPI_TABLE_GUID.
//
#define EFI_ACPI_20_TABLE_GUID EFI_ACPI_TABLE_GUID

extern EFI_GUID gEfiAcpiTableGuid;
extern EFI_GUID gEfiAcpi10TableGuid;
extern EFI_GUID gEfiAcpi20TableGuid;
```

ACPI table [illegible]

[illegible] 1.0 [illegible] 2.0 [illegible]

image-20240729163134162

image-20240729103538772

image-20240729103136631

[illegible] ACPI table entry [illegible] RSDP [illegible]RSDP [illegible] XSDT[illegible]

RSDT [illegible] XSDT [illegible]

[illegible]

- RSDT [illegible] 32 [illegible] 4GB [illegible] ACPI [illegible]
- XSDT [illegible] 64 [illegible] 64 [illegible]

[illegible]

- RSDT [illegible] ACPI 1.0 [illegible] 32 [illegible]
- XSDT [illegible] ACPI 2.0 [illegible] 64 [illegible] 32 [illegible]

[illegible]

- RSDT [illegible] 32 [illegible]
- XSDT [illegible] 64 [illegible]

Linux Kernel [illegible] ACPI

[illegible] ACPI [illegible]

[illegible] ACPI table

[illegible] ACPI table[illegible] XSDT [illegible] 36 [illegible] entry[illegible] RSDP[illegible]XSDT[illegible]FADT[illegible]Fixed ACPI Description Table[illegible] DSDT[illegible]Differentiated System Description Table[illegible]FACS[illegible]Firmware ACPI Control Structure[illegible]

```
Signature "XSDT"
Length  0x00000144 (324)
Revision  0x01 (1)
Checksum  0x34 (52)
OEM ID  "LENOVO"
OEM Table ID  "CB-01   "
OEM Revision  0x00000001 (1)
Creator ID  "    "
Creator Revision  0x01000013 (16777235)
Entry0  0x0000000044BC3000 (FACP)
Entry1  0x0000000044B2E000 (UEFI)
Entry2  0x0000000044BF3000 (SSDT)
Entry3  0x0000000044BF2000 (SSDT)
Entry4  0x0000000044BEC000 (SSDT)
Entry5  0x0000000044BE8000 (SSDT)
Entry6  0x0000000044BE4000 (SSDT)
Entry7  0x0000000044BD6000 (SSDT)
Entry8  0x0000000044BD5000 (SSDT)
Entry9  0x0000000044BD4000 (TPM2)
Entry10 0x0000000044BD3000 (SSDT)
Entry11 0x0000000044BD2000 (SSDT)
Entry12 0x0000000044BD1000 (MSDM)
Entry13 0x0000000044BCF000 (SSDT)
Entry14 0x0000000044BCE000 (LPIT)
Entry15 0x0000000044BCD000 (WSMT)
Entry16 0x0000000044BCC000 (SSDT)
Entry17 0x0000000044BC9000 (SSDT)
Entry18 0x0000000044BC8000 (DBGP)
Entry19 0x0000000044BC7000 (DBG2)
Entry20 0x0000000044BC4000 (NHLT)
Entry21 0x0000000044BFD000 (ECDT)
Entry22 0x0000000044BC2000 (HPET)
Entry23 0x0000000044BC1000 (APIC)
Entry24 0x0000000044BC0000 (MCFG)
Entry25 0x0000000044B58000 (SSDT)
Entry26 0x0000000044B56000 (SSDT)
Entry27 0x0000000044B55000 ($H2O)
Entry28 0x0000000044B54000 (DMAR)
Entry29 0x0000000044B53000 (SSDT)
Entry30 0x0000000044B4F000 (SSDT)
Entry31 0x0000000044B4B000 (SSDT)
```

```
Entry32 0x0000000044B4A000 (SSDT)
Entry33 0x0000000044B49000 (FPDT)
Entry34 0x0000000044B48000 (BGRT)
Entry35 0x0000000044B47000 (PHAT)
```

RSDP[illegible]Root System Description Pointer[illegible]

- [illegible] RSDT [illegible] XSDT [illegible]
- [illegible]

RSDT[illegible]Root System Description Table[illegible] XSDT[illegible]Extended System Description Table[illegible]

- [illegible] ACPI [illegible]
- RSDT [illegible] 32 [illegible]XSDT [illegible] 64 [illegible]

FADT[illegible]Fixed ACPI Description Table[illegible]

- [illegible]
- [illegible]

DSDT[illegible]Differentiated System Description Table[illegible]

- [illegible]AML [illegible]
- [illegible]

FACS[illegible]Firmware ACPI Control Structure[illegible]

- [illegible]
- [illegible]

[illegible] Memory [illegible] ACPI table

[illegible] ACPI [illegible] `SRAT`[illegible]System Resource Affinity Table[illegible] `SLIT`[illegible]System Locality Information Table[illegible]

MCFG[illegible]Memory Configuration Table[illegible]

- MCFG [illegible] ACPI [illegible] DRAM[illegible]Dynamic Random Access Memory[illegible] MMIO[illegible]Memory-Mapped I/O[illegible]/[illegible] MCFG [illegible] DRAM[illegible] MMIO[illegible]

SRAT[illegible]System Resource Affinity Table[illegible]

`SRAT` [illegible]CPU [illegible] I/O [illegible] NUMA[illegible]Non-Uniform Memory Access[illegible]`SRAT` [illegible]

- [illegible]
- [illegible]
- [illegible] CPU [illegible]
- I/O [illegible]

SRAT [illegible]

```
struct ACPI_TABLE_SRAT {
    struct ACPI_TABLE_HEADER Header;      // ACPI [illegible]
    uint32_t Reserved1;                   // [illegible]
    uint64_t Reserved2;                   // [illegible]
    struct ACPI_SRAT_ENTRY Entries[];     // [illegible]
};

struct ACPI_SRAT_ENTRY {
    uint8_t Type;                         // [illegible] CPU[illegible]
    uint8_t Length;                       // [illegible]
    // [illegible] Type [illegible]
};

// [illegible]
struct ACPI_SRAT_MEM_AFFINITY {
    uint8_t Type;                         // [illegible]0x01 [illegible]
    uint8_t Length;                       // [illegible]
    uint32_t ProximityDomain;             // [illegible]
    uint16_t Reserved1;                   // [illegible]
    uint16_t Flags;                       // [illegible]
    uint64_t BaseAddress;                 // [illegible]
    uint64_t Length;                      // [illegible]
    uint32_t Reserved2;                   // [illegible]
    uint32_t Reserved3;                   // [illegible]
};
```

SLIT[illegible]System Locality Information Table[illegible]

SLIT [illegible]SLIT [illegible]

```
struct ACPI_TABLE_SLIT {
    struct ACPI_TABLE_HEADER Header;      // ACPI [illegible]
    uint64_t LocalityCount;               // [illegible]
    uint8_t Entry[];                      // [illegible]
};
```

[illegible] CXL [illegible] ACPI table

CEDT [illegible] ACPI [illegible] CXL Early Discovery Table[illegible]CXL [illegible]

CEDT [illegible] CXL [illegible] CXL [illegible]

CEDT [illegible]

CEDT [illegible] CXL [illegible] CXL [illegible] CXL [illegible]

CEDT [illegible]

CEDT [illegible] CXL [illegible]Entries[illegible]

1. **CXL Host Bridge Structure**[illegible] CXL [illegible]
2. **CXL Device Structure**[illegible] CXL [illegible] ID[illegible]
3. **CXL Switch Structure**[illegible] CXL [illegible]Switch[illegible]

[illegible] CXL [illegible] CXL [illegible]

CEDT [illegible]

```c
struct cedt_header {
    uint32_t signature;    // 'CEDT'
    uint32_t length;       // Length of the entire table
    uint8_t revision;      // Revision of the CEDT table
    uint8_t checksum;      // Checksum of the entire table
    char oem_id[6];        // OEM ID
    char oem_table_id[8];  // OEM Table ID
    uint32_t oem_revision; // OEM Revision
    uint32_t creator_id;   // Creator ID
    uint32_t creator_rev;  // Creator Revision
};

struct cxl_host_bridge {
    uint16_t type;         // Entry type (CXL Host Bridge)
    uint16_t length;       // Length of this entry
    uint32_t host_bridge_id; // Host Bridge ID
    // Additional fields specific to the CXL Host Bridge
};

struct cxl_device {
    uint16_t type;         // Entry type (CXL Device)
    uint16_t length;       // Length of this entry
    uint32_t device_id;    // Device ID
    // Additional fields specific to the CXL Device
};

// Main CEDT Table containing multiple entries
struct cedt_table {
    struct cedt_header header;
    struct cxl_host_bridge host_bridge;
    struct cxl_device device;
    // Other entries as needed
};
```

□□□

□□□□□□ DSDT □ SSDT □□

ACPI □□□ namespace□□□□□□□

image-20240729161807074

image-20240729161816008

□□□□□ table

asl □□□ aml □□□aml □□□ raw□raw □□□ ffs

□ ffs □□□ aml □□□□□ memory □□□□□□ protocol □ install□

```
// Example SSDT: SSDT_MYCUSTOM.aml

DefinitionBlock ("SSDT_MYCUSTOM.aml", "SSDT", 1, "YOURID", "YOURID", 1)
{
    // Insert your ACPI table content here
    // Example:
    Method (_STA, 0, NotSerialized)  // _STA: Status
    {
        Return (0x0F) // indicate device is present and working
    }
}
```

1. **DefinitionBlock**□- `DefinitionBlock` □□□□ ACPI □□□□□□□□□□□□□□□□`SSDT_MYCUSTOM.aml`□□□□□□`SSDT` □□ Secondary System Description Table□□ACPI □□□□□`1`□□OEM ID□`"YOURID"`□□OEM □□ ID□`"YOURID"`□□
2. **Method (_STA)**□- □ ACPI □□□□□□□□□□□□□□□□□□□`Method (_STA, 0, NotSerialized)` □□□□□□□ `_STA` □□□□□□□□□`0`□□□□□□□□□□□□`NotSerialized`□□

- `_STA` □□□□□□□□□□□□□□□□□□□`Return (0x0F)` □□□□□□□□□□□`0x0F`□□□□□□□□□□□□□□□□□

EDK2 protocol

Status = gBS->LocateProtocol (&gEfiAcpiTableProtocolGuid, NULL, (VOID **) &AcpiTable);

Status = AcpiTable->InstallAcpiTable (AcpiTable, Table, TableSize, &TableKey); □□□ uninstall□□□□□ ACPI table□

□□□□□□□ table□□□□□□□□□□□□□□□□□□□ device tree □□□

□□

https://www.nirsoft.net/utils/firmware _tables _view.html

RW

UEFI Shell ☒ https://acpica.org/downloads/uefi-support

acpi windwos tool download link: `https://acpica.org/downloads/binary-tools`

UEFI Shell ☒ https://github.com/andreiw/UefiToolsPkg/tree/master/Applications

UEFI Shell ☒☒☒ acpiview from UefiShellAcpiViewCommandLib.inf

☒☒☒

Advanced Configuration and Power Interface (ACPI) Introduction and Overview

☒☒☒☒☒☒☒☒☒(ACPI)☒☒

ACPI Spec

https://www.lab-z.com/revmem/

ACPI Source Language (ASL) Tutorial

CEDT

[toc]

CEDT [illegible] ACPI [illegible] CXL Early Discovery Table[illegible]CXL [illegible]

CEDT [illegible] CXL [illegible] CXL [illegible]

CEDT [illegible]

CEDT [illegible] CXL [illegible] CXL [illegible] CXL [illegible]

CEDT [illegible]

CEDT [illegible] CXL [illegible]Entries[illegible]

1. **CXL Host Bridge Structure**[illegible] CXL [illegible]
2. **CXL Device Structure**[illegible] CXL [illegible] ID[illegible]
3. **CXL Switch Structure**[illegible] CXL [illegible]Switch[illegible]

[illegible] CXL [illegible] CXL [illegible]

CEDT [illegible]

image-20240729154011756

image-20240729154518068

```
#define INTERLEAVE_TARGETS  1

#pragma pack (1)

typedef struct {
  EFI_ACPI_CEDT_CFMWS_STRUCTURE  Cfmws;
  UINT32  InterleaveTarget[INTERLEAVE_TARGETS];
} EFI_ACPI_CEDT_CFMWS_AND_INTERLEAVE_TARGET_STRUCTURE;

typedef struct {
  EFI_ACPI_DESCRIPTION_HEADER  Header;
  EFI_ACPI_CEDT_CFMWS_AND_INTERLEAVE_TARGET_STRUCTURE CfmwsTarget;
  EFI_ACPI_CEDT_CHBS_STRUCTURE  Chbs;
} EFI_ACPI_CEDT_STRUCTURE_TABLE;

#pragma pack ()

STATIC EFI_ACPI_CEDT_STRUCTURE_TABLE Cedt = {
  ARM_ACPI_HEADER (
    EFI_ACPI_6_4_CEDT_SIGNATURE,
    EFI_ACPI_CEDT_STRUCTURE_TABLE,
    1
  ),
  {
    {
    // CFMWS
      {
        1,
        EFI_ACPI_RESERVED_BYTE,
        sizeof (EFI_ACPI_CEDT_CFMWS_AND_INTERLEAVE_TARGET_STRUCTURE)
      },
      EFI_ACPI_RESERVED_DWORD,
      0x3fe00000000,
      0x200000000,
      0,
      0,
      EFI_ACPI_RESERVED_WORD,
      4,
      2,
      0
    },
    {
    //Interleave target list
      1
    }
  },
  {
  // CHBS
    {
      0,
      EFI_ACPI_RESERVED_BYTE,
      sizeof (EFI_ACPI_CEDT_CHBS_STRUCTURE)
    },
    1,
    1,
    EFI_ACPI_RESERVED_DWORD,
    0x10D0000000,
    0x1000
  }
};

//
// Reference the table being generated to prevent the optimizer from removing
// the data structure from the executable
//
VOID* CONST ReferenceAcpiTable = &Cedt;
```

☒☒☒

CXL Type-3 device discovery, configuration in firmware and prepare ACPI tables for kernel usage

EDK2 CEDT Table

CXL Spec3.0 Chapter 9.17

AMBA

PCIe

PCIe

Physical Layer

[illegible]Link [illegible] Lane[illegible]EQ[illegible]LTSSM

Introduction

[illegible]

image-20240704095625749

[illegible]

- [illegible]Logical Sub-Block[illegible]
- [illegible]Electrical Sub-Block[illegible]

Logical Sub-Block

image-20240705171230174

Tx Buffer & Mux

[illegible] buffer[illegible] DL [illegible] Mux[illegible] DL [illegible] skip [illegible] buffer[illegible]

Byte Striping

[illegible] lane [illegible]

image-20240705172339842

Data Scrambling

[illegible]Scrambling[illegible]pseudo-random number generator[illegible]

[illegible] EMI[illegible] 0 [illegible] 1[illegible] 01 [illegible]Scrambling[illegible]

[illegible] PCIe [illegible] —— [illegible] **XOR** [illegible]**LFSR**[illegible]**Linear-feedback shift register**[illegible]

image-20240705172450492

PCIe [illegible]Galois[illegible] LFSR[illegible] PCIe 1.0 [illegible] 2.0 [illegible] 16 [illegible] LFSR [illegible]

image-20240705172531352

[illegible] 3.0 [illegible] 23 [illegible] LFSR[illegible]

image-20240705172536567

[illegible]LFSR [illegible] bit[illegible] bit [illegible] XOR [illegible]

pcie-phy-lfsr-galois

[illegible]PCIe1.0 [illegible] 2.0 [illegible] LFSR [illegible]Seed[illegible] 0xFFFF[illegible] PCIe3.0 [illegible] Lane [illegible] Lane [illegible] LFSR [illegible]Seed[illegible] 8 [illegible] Lane ID [illegible] 8 [illegible]

Lane	Seed
0	1DBFBCh
1	0607BBh
2	1EC760h
3	18C0DBh
4	010F12h
5	19CFC9h
6	0277CEh
7	1BB807h

[illegible] XOR [illegible] XOR [illegible] Seed [illegible] LFSR [illegible]

[illegible]

Encoding

[illegible]Encoding[illegible] 0 [illegible] 1 [illegible] DC [illegible]DC Balance[illegible] PCIe [illegible]Clock Recovery[illegible]

PCIe [illegible]8b/10b [illegible]128b/130b [illegible] 242B/256B FLIT [illegible] PCIe [illegible] Lane [illegible] PCIe 1.0 8b/10b [illegible] 8bit [illegible] 10b [illegible] Lane [illegible]

image-20240705173724380

[illegible] PCIe [illegible]

PCIe Version	Line code	Transfer rate per lane	Throughput x1	Throughput x16
1.0	8b/10b	2.5 GT/s	250 MB/s	4 GB/s
2.0	8b/10b	5 GT/s	500 MB/s	8 GB/s
3.0	128b/130b	8 GT/s	984.6 MB/s	15.75 GB/s
4.0	128b/130b	16 GT/s	1.969 GB/s	31.51 GB/s
5.0	128b/130b	32 GT/s	3.938 GB/s	63.02 GB/s
6.0	1b/1b 242B/256B FLIT	64 GT/s	7.564 GB/s	121.00 GB/s

[illegible]242B/256B [illegible] B[illegible] bit [illegible] Byte [illegible]

8b/10b [illegible]

8b/10b [illegible] PCIe 1.0 [illegible] 2.0 [illegible] 2.5GT/s [illegible] 5GT/s [illegible] [4][illegible] 8bits [illegible] 5bits [illegible] 3bits [illegible] 6bits [illegible] 4bits [illegible] 0 [illegible] 1 [illegible]`ABCDEFGH` [illegible] `ABCDE` [illegible] `FGH`[illegible] `abcdei` [illegible] `fghj` [illegible]

[illegible] 8b/10b [illegible]K[illegible]D[illegible] `ABCDE` [illegible] `FGH` [illegible] `D.<ABCDE>.<FGH>`[illegible] `K.<ABCDE>.<FGH>`[illegible]

image-20240705174102857

image-20240705174110853

☒☒**RD☒Running Disparity☒**☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ 1 ☒☒☒☒☒ 0 ☒☒☒☒☒☒☒☒☒☒☒☒☒ 01 ☒☒☒☒☒8b/10b ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒RD ☒☒☒☒☒☒±1☒☒☒☒☒☒☒☒☒RD ☒☒☒☒☒☒±2☒

☒☒☒☒☒☒☒☒☒8b/10b ☒☒☒☒☒☒ 01 ☒☒☒☒☒☒☒☒☒☒☒☒☒ DC ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ 20% ☒☒☒☒☒☒☒☒☒☒ PCIe 3.0 ☒☒☒☒☒☒☒ 8b/10b ☒☒☒☒☒☒

128b/130b ☒☒

PCIe 3.0 ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒PCIe ☒☒☒☒ 128b/130b ☒☒☒☒☒☒☒☒☒ 130 ☒ bits ☒☒☒☒ 2 ☒ bits ☒☒☒☒☒☒☒☒☒☒☒☒☒ 98.46%☒

128b/130b ☒☒☒ 64b/66b ☒☒☒☒☒ [5]☒☒☒☒☒☒☒☒☒ payload ☒☒☒☒☒☒☒☒☒64b -> 128b☒☒☒ 8b/10b ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ 3.0 ☒☒☒ LFSR ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ DC ☒☒☒☒☒☒☒

☒☒☒☒ LFSR ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒128b/130b ☒☒☒☒☒☒☒☒☒Framing ☒ Encoding☒

Framing

☒☒☒128b/130b ☒☒☒☒☒☒☒☒☒☒☒☒☒☒ block☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ Token☒☒☒☒☒☒☒☒☒☒☒

- ☒☒ TLP ☒☒☒☒☒☒☒☒☒☒☒ 2 ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ 4 ☒ bits☒☒☒☒☒☒☒ FCRC☒☒☒☒☒☒☒☒☒☒☒☒ Seq ☒☒☒☒☒☒ 4 ☒☒☒ token —— STP☒Start of TLP☒☒

image-20240705174145453

- ☒☒ DLLP ☒☒☒☒☒☒☒☒ 2 ☒☒☒ Token - SDP☒Start of DLLP☒☒☒☒☒☒

image-20240705174151441

☒☒☒ Token ☒☒☒☒☒ Lane 0 ☒☒☒☒☒☒☒☒☒☒☒☒☒☒Byte 0☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ IDL☒Logical Idle☒Nullify ☒☒ TLP ☒ EDB☒EnD Bad☒☒☒☒☒☒☒☒☒☒ EDS☒End of Data Stream☒☒

image-20240705174208546

Encoding

☒☒☒128b/130b ☒☒☒☒☒☒☒☒☒☒☒☒☒ 128bit ☒ block☒☒☒☒☒ 128bits ☒ payload ☒☒☒☒☒☒ 2bits ☒☒☒☒☒SyncHeader☒☒☒☒☒ payload ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

- 01b☒☒☒☒☒ payload ☒☒☒☒☒Data Block☒☒☒☒ 128bits
- 10b☒☒☒☒☒ payload ☒ Ordered Set Block☒☒☒☒☒ 128bits

☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ 16 ☒☒☒☒S0-S15☒☒☒☒☒☒ bit ☒☒☒☒.0☒☒☒☒☒☒.7☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ 01b ☒☒☒ 10b☒☒

image-20240705174235248

☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

image-20240705174247660

242B/256B FLIT [illegible]

[illegible] 128b/130b [illegible]PCIe6.0 [illegible] PCIe1.0 [illegible] NRZ[illegible]Non-Return-To-Zero[illegible] PAM4[illegible]Pulse Amplitude Modulation 4[illegible] PCIe6.0 [illegible]PCIe [illegible] 242B/256B FLIT [illegible]

242B/256B FLIT [illegible] FLIT[illegible] 256 [illegible] bits[illegible]

image-20240705174353371

[illegible] 236 [illegible] TLP [illegible] 6 [illegible] DLLP[illegible]Data Link Layer Packet[illegible]8 [illegible] CRC[illegible] 6 [illegible] FEC[illegible]Forward Error Correction[illegible] FLIT [illegible] CRC [illegible] DLLP [illegible] TLP [illegible] CRC [illegible] FEC [illegible] FLIT [illegible]

[illegible] FLIT [illegible]PCIe [illegible] FLIT [illegible] NRZ [illegible] 2.5 GT/s[illegible]5.0 GT/s[illegible]8.0 GT/s[illegible]16.0 GT/s[illegible] 32.0 GT/s[illegible] NRZ [illegible]

Electrical Sub-block

[illegible]Parallel to Serial[illegible]

[illegible]

[illegible]Modulation[illegible]

[illegible]

NRZ [illegible]Non-Return-to-Zero[illegible]

[illegible] PCIe1.0 [illegible] 5.0 [illegible]PCIe [illegible]NRZ[illegible]Non-Return-to-Zero[illegible] [10][illegible]RZ[illegible] Return-to-Zero[illegible] 1 [illegible]PCIe [illegible]Bipolar NRZ level[illegible] 0 [illegible]−V[illegible]1 [illegible]+V[illegible]

image-20240705174549147

[illegible]Encoding[illegible]Scrambling[illegible]

PAM4 [illegible]

[illegible]PCIe6.0 [illegible] PAM4[illegible]Pulse Amplitude Modulation 4[illegible] 00 [illegible]−V[illegible]10 [illegible]+V[illegible] 01 [illegible]−V/3[illegible]11 [illegible]+V/3[illegible]

image-20240705174638258

[illegible]

[illegible]Differential Signal[illegible]

[illegible]PCIe [illegible]Differential Signaling[illegible] Pinout [illegible] Differential Pair[illegible]

image-20240705174714897

image-20240705174727675

image-20240705174944032

[illegible] Blog

Data Link Layer

Introduction

[illegible] TLP[illegible]Ack/Nack[illegible]

[illegible]Data Link Layer[illegible]

[illegible] **TLP** [illegible]Feature[illegible] ack/nack [illegible] Token [illegible]STP[illegible]Start of TLP[illegible] TLP [illegible]SDP[illegible]Start of DLLP[illegible]DLLP[illegible]Data Link Layer Packet[illegible]

[illegible] Transaction layer [illegible] TLP [illegible] TLP [illegible]buffer [illegible] TLP [illegible]

image-20240705142038727

image-20240705142914955

[illegible] TLP

[illegible]

1. [illegible]Sequence Number[illegible] 2 [illegible] Link [illegible] `[illegible]NEXT_RCV_SEQ[illegible]`
2. [illegible] CRC [illegible] LCRC[illegible]Link CRC[illegible] 4 [illegible] CRC [illegible]

image-20240705144307306

[illegible]

[illegible] Retry Buffer [illegible]Physical Layer[illegible] ACK [illegible] Seq [illegible]CRC [illegible] Retry Buffer [illegible]

[illegible]

[illegible] CRC [illegible] Nak [illegible] Ack [illegible] ACK [illegible] Retry Buffer [illegible] TLP [illegible]

[illegible] DLLP

[illegible] TLP [illegible] Ack [illegible] Nak[illegible] DLLP[illegible]Data Link Layer Packet[illegible] 8B[illegible]DLLP [illegible] 6B[illegible] 2B [illegible](gen1&2 [illegible] SDP [illegible] end[illegible]gen3 [illegible] 2B [illegible] SDP)[illegible] 8B[illegible]

[illegible]

image-20240705144504223

image-20240705153319533

image-20240705153333401

[illegible] TLP [illegible] DLL [illegible] CRC [illegible] 32bit[illegible] DLLP [illegible] CRC [illegible] 16bit[illegible]

DLLP [illegible] DLLP Type [illegible] 16 [illegible] CRC [illegible]

[illegible]	Type	[illegible]
Ack	00000000b	[illegible] TLP [illegible]
Nak	00010000	[illegible] TLP [illegible]
<InitFC1/InitFC2/UpdateFC>-<P/NP/Cpl>	[illegible]Type [illegible]	[illegible]P/NP/Cpl [illegible]
MRInitFC1/MRInitFC2/MRUpdateFC	<0111/1111/1011>0xxxb	[illegible]P/NP/Cpl [illegible]
PM _*	00100xxxb	[illegible]
NOP	00110001b	[illegible]
Data _Link _Feature	00000010b	[illegible] Scaled Flow Control
Vendor-specific	00110000b	[illegible] DLLP[illegible]

Ack/Nak

[illegible] TLP [illegible] Ack [illegible] Nak [illegible] DLLP [illegible]Ack [illegible]Nak [illegible]

image-20240705153917566

[illegible]AckNak _Seq _Num [illegible] TCP [illegible]PCIe [illegible] Ack [illegible] Nak [illegible] Ack [illegible] Nak[illegible] Retry Buffer [illegible] Ack/Nak [illegible]Ack/Nak [illegible] Nak[illegible] Retry Buffer [illegible]

[illegible]DDLP [illegible] 4 [illegible]retrain[illegible]

VC[illegible]Virtual Channel[illegible]

[illegible] TLP [illegible] PCIe [illegible] TC[illegible]Traffic Class[illegible] VC[illegible]Virtual Channel[illegible] VC [illegible]

[illegible]

1. [illegible] TLP [illegible]Posted[illegible]P[illegible]Non-Posted[illegible]NP[illegible] Completion[illegible]Cpl[illegible]
2. [illegible] VC [illegible] Link[illegible] Link [illegible] VC[illegible] VC [illegible]

[illegible]N/NP/Cpl[illegible]

- **InitFC1-P/NP/Cpl**[illegible]
- **InitFC2-P/NP/Cpl**[illegible] InitFC1 [illegible] InitFC1 [illegible]
- **UpdateFC-P/NP/Cpl**[illegible]

image-20240705153948650

[illegible]

- **Type**[illegible] ID[illegible]

| Type | Id | | ——— | —– | | InitFC1-P | 0100b | | InitFC1-NP | 0101b | | InitFC1-Cpl | 0110b | | InitFC2-P | 1100b | | InitFC2-NP | 1101b | | InitFC2-Cpl | 1110b | | UpdateFC-P | 1000b | | UpdateFC-NP | 1001b | | UpdateFC-Cpl | 1010b |

- **VC ID**[illegible]**v[2:0]**[illegible]Virtual Channel [illegible] Id[illegible]Id [illegible] 3 [illegible] 8 [illegible] VC[illegible]
- **HdrFC**[illegible]TLP [illegible] Credit [illegible] TLP [illegible] Header Credit[illegible] TLP [illegible]
- **DataFC**[illegible]TLP [illegible] Credit [illegible] DW[illegible]Double Word[illegible] 4 [illegible] Data Credit[illegible]

[illegible] 64 [illegible] 128 [illegible] 4 DW [illegible] TLP [illegible] 128 [illegible] Payload[illegible] 1 DW [illegible] TLP Digest[illegible] 1 [illegible] Header Credit[illegible] (128 + 4) / 4 = 33 [illegible] Data Credit[illegible]

[illegible] UpdateFC [illegible] 30us[illegible] CRC [illegible]

[illegible] Scaled Flow Control[illegible] 2 [illegible]

image-20240705154129709

AVIP log [illegible] credit [illegible]6 [illegible] stack[illegible]post non-post cpl head and data

image-20240705154107474

[illegible]

[illegible]HdrFC [illegible] DataFC [illegible]

1. [illegible]PCIe [illegible] Endpoint [illegible] Switch [illegible]

image-20240705154329341

1. [illegible] Switch [illegible] Endpoint [illegible] Switch [illegible] Endpoint [illegible]
2. Switch [illegible] InitFC1 DLLP [illegible] InitFC2 DLLP [illegible]

image-20240705154350541

[illegible][Blog](Blog)

Transaction Layer

[toc]

[illegible]VC[illegible]Ordering

Introduction

PCIe [illegible]Transaction[illegible]

- [illegible]Memory Transaction[illegible]
- IO [illegible]IO Transaction[illegible]
- [illegible]Configuration Transaction[illegible]
- [illegible]Message Transaction[illegible]

[illegible]

- **Non-Posted**[illegible]Completion[illegible]
- **Posted**[illegible] Fire and forget[illegible]

[illegible]Non-Posted[illegible]NP[illegible]Posted[illegible]P[illegible] Completion[illegible]Cpl[illegible]

image-20240705142038727

TLP[illegible]Transaction Layer Packet[illegible]

PCIe [illegible] TLP[illegible]Transaction Layer Packet[illegible]

image-20240705140042775

- **TLP Prefix**[illegible]Precision Time Measurement[illegible]

SMMU [illegible] substreamID [illegible] PASID[illegible]

- **TLP Digest**[illegible]4 [illegible] CRC [illegible]

[illegible] DLLP [illegible] CRC[illegible] CRC[illegible] switch [illegible] DLLP [illegible] CRC[illegible] TLP [illegible] switch [illegible] TLP [illegible] TLP [illegible]

- **TLP Header**[illegible] TLP [illegible]
- **TLP Payload**[illegible] TLP [illegible] Payload[illegible] Payload [illegible] `Max_Payload_Size`[illegible] 4096 [illegible]

[illegible] buffer [illegible] buffer [illegible] `Max_Payload_Size`

TLP [illegible]

TLP [illegible] 12 [illegible] 3DW[illegible] 16 [illegible] 4DW[illegible] 4 [illegible] DW[illegible] 8 [illegible] DW[illegible]

image-20240705140608530

- **Fmt**

: TLP [illegible]

- **Bit 7**[illegible] 1[illegible] Fmt [illegible] 100[illegible] TLP Prefix
- **Bit 6**[illegible]1 = [illegible]TLP [illegible] Payload[illegible]0 = [illegible]TLP [illegible] Payload[illegible]
- **Bit 5**[illegible]1 = [illegible] 32 [illegible] 12 [illegible]3DW Header[illegible]0 = [illegible] 64 [illegible] 16 [illegible]4DW Header[illegible]
- **Type**[illegible]IO [illegible]

[illegible] TLP [illegible]

- **LN**[illegible]**Lightweight Notification**[illegible]
- **TH**[illegible]**TLP Hints**[illegible] TPH[illegible]TLP Processing Hint[illegible] TPH TLP Prefix [illegible]
- **TD**[illegible]**TLP Digest**[illegible]1 = [illegible] TLP Digest[illegible]0 = [illegible] TLP Digest
- **EP**[illegible]**Error Poisoning**[illegible]1 = [illegible]0 = [illegible]
- **AT**[illegible]**Address Type**[illegible]ATS [illegible]00 = [illegible]01 = [illegible]10 = [illegible]11 = [illegible]
- **Length**[illegible]Payload [illegible] **DW**[illegible]**Double Word**[illegible]1DW = 4 [illegible]

[illegible]Transaction Descriptor[illegible]

[illegible] TLP [illegible] ID[illegible]Requester ID [illegible] Tag [illegible]Attr [illegible]TC [illegible] TLP [illegible] DW [illegible] Tag [illegible] TLP [illegible]

image-20240705140840954

[illegible] ID[illegible]Transaction ID[illegible]

[illegible] ID [illegible] Requester ID [illegible] Tag [illegible]

- **Requester ID**[illegible] 16 [illegible] bit[illegible] BDF[illegible] 8:5:3.
- **Tag**[illegible] 10 [illegible] bit[illegible] TLP [illegible] PCIe [illegible] T8 [illegible] T9 [illegible] bits[illegible] tag [illegible] bits [illegible] `10-Bit Tag Requester Enable` [illegible]

[illegible]Attributes[illegible]

[illegible] bits[illegible] Attr[2:1][illegible]Byte 1 - Bit 2[illegible]Byte 2 - Bit 5[illegible] Attr[0][illegible]Byte 2 - Bit 4[illegible] Coherency[illegible]

image-20240705141055372

[illegible]Ordering[illegible]

PCIe [illegible] virtual channel[illegible] TLP [illegible] TC[illegible] buffer [illegible] VC [illegible] VC [illegible] TC TLP [illegible] ordering [illegible]

image-20240705141430718

Attr[2:1][illegible] Bits [illegible]

| Attr[2] | Attr[1] | [illegible] | [illegible] | | —– | —– | ———— | ———————————————— | | 0 | 0 | [illegible] | [illegible] | | 0 | 1 | Relaxed Ordering | [illegible] | | 1 | 0 | ID-based Ordering | [illegible] | | 1 | 1 | [illegible] | [illegible] Relaxed Ordering [illegible] ID-based Ordering [illegible] |

No Snoop

NoSnoop[illegible]Attr[0][illegible] 0[illegible]PCIe [illegible] Cache[illegible] 1[illegible]PCIe [illegible] Cache[illegible] Cache [illegible]

[illegible] flag [illegible] flag[illegible]

[illegible]IO [illegible] MSI[illegible] DMA [illegible]

[illegible]Traffic Class[illegible]

Traffic Class [illegible] 3 [illegible] bit[illegible] 8 [illegible]

[illegible] TC [illegible] VC[illegible]Virtual Channel[illegible]

1. PCIe [illegible]Link[illegible] VC[illegible]Virtual Channel[illegible] VC [illegible]
2. [illegible] TC [illegible] VC [illegible] TLP [illegible] TC [illegible] TLP [illegible] VC [illegible]
3. VC [illegible] VC [illegible] Credit Pool[illegible] VC [illegible] Credit [illegible] 0[illegible] TLP[illegible] Credit[illegible] VC [illegible] Credit [illegible]
4. TC [illegible] 0[illegible] Hardcoded [illegible] VC0 [illegible] TC[illegible] TLP [illegible] VC0[illegible]
5. [illegible] VC[illegible] TC[illegible]

TLP [illegible]

[illegible] ID[illegible]

[illegible] PCIe [illegible]

[illegible]**Address-Based Routing**[illegible]Memory Transaction[illegible] IO [illegible]IO Transaction[illegible]

image-20240705141753898

[illegible] **BDF** [illegible] **ID** [illegible]**ID Based Routing**[illegible]Configuration Transaction[illegible]Message Transaction[illegible]Completion[illegible]

[illegible]

Implicit routing is used only with **Message Requests**, and is covered in Section 2.2.8

PCIe [illegible]

[toc]

Overview

[illegible]

[illegible]([illegible])[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]De-emphasis[illegible] Pre-shoot [illegible]

[illegible]

[illegible]Feed forward Equalizer[illegible]FFE[illegible]

[illegible]

[illegible] 111101111 [illegible] **0**[illegible] **0**[illegible] **1**[illegible] **0** [illegible]

image-20240726102421538

[illegible] **1** [illegible] **0** [illegible]

image-20240726102525237

1. [illegible]
2. [illegible]

[illegible]

[illegible] Preshoot [illegible] De-emphasis [illegible]

image-20240726102648954

[illegible] FIR

8 GT/s [illegible]Finite Impulse Response, FIR[illegible]

[illegible] FIR [illegible] 5 [illegible]

image-20240726105305332

[illegible] C-1[illegible]C0[illegible]C+1 [illegible] FIR [illegible] Pre-cursor[illegible]Cusor [illegible] Post-cursor [illegible]Pre-cursor [illegible] [illegible] Pre-shoot [illegible]Post-cursor [illegible]Pre-cursor[illegible]Cusor [illegible] Post-cursor [illegible] image-20240726105334155

[illegible] Pre-cursor [illegible] Post-cursor [illegible]Cursor [illegible]

image-20240726105353700

[illegible] de-emphasize [illegible] preshoot [illegible]

[illegible] preset [illegible]

[illegible]

image-20240726105914739

[illegible]

[illegible]

[illegible]Continuous-Time Linear Equalizaion, CTLE[illegible]Decision Feedback Equalization[illegible]DFE[illegible]

CTLE

[illegible]

DFE

DFE [illegible]

DFE [illegible]

DFE [illegible]

[illegible]

[illegible]-3.5dB[illegible]

5 GT/s [illegible]

[illegible]-3.5dB [illegible]-6dB[illegible]

8 GT/s[illegible]16 GT/s[illegible]32 GT/s [illegible]

gen1, gen2 [illegible]

gen3 [illegible] FIR[illegible] CTLE [illegible] DFE[illegible]

[illegible]

[illegible]PCIe [illegible]

[illegible] PCIe [illegible]

[illegible] 8 GT/s [illegible] 16 GT/s [illegible] 32 GT/s [illegible]

[illegible] L0 [illegible] PCIe [illegible] EQ [illegible] EQ[illegible] DLLP[illegible]

[illegible]

- [illegible] Link Control 3 Register [illegible] Perform Equalization [illegible]
- [illegible] Link Control 2 Register [illegible] Target Link Speed [illegible] 8 GT/s [illegible]
- [illegible] Link Control Register [illegible] Retrain Link [illegible]

[illegible] DLLP Blocking [illegible]

[illegible]

Full equalization mode[illegible] [illegible] PCIe [illegible]

[illegible] training[illegible]

[illegible] PCIe [illegible] EQ [illegible] 100 ms[illegible]EQ [illegible] PCIe Gen5 [illegible]PCIe [illegible]

[illegible]

[illegible] LTSSM Configuration [illegible] TS [illegible] Equalization bypass to highest rate[illegible]

[illegible] 32GT Capability Reg [illegible]Equalization bypass to highest rate

[illegible] Bypass equalization to highest rate [illegible] 2.5 GT/s L0 [illegible] Recovery [illegible]32 GT/s [illegible] 1 [illegible] EQ[illegible]

[illegible]

[illegible] 32 GT/s [illegible] No equalization needed[illegible] 5 GT/s [illegible] EQ [illegible] EQ [illegible] EQ [illegible]

Gen1 [illegible] gen5[illegible]

[illegible]

[illegible] Phase[illegible]PCIe Controller [illegible]link status[illegible] Equalization Phase Successful [illegible]

[illegible] Phase 3 [illegible] Equalization Complete [illegible]

8GT [illegible] link status

16 [illegible] 32GT [illegible] 16 [illegible] 32GT status

[illegible]

[illegible] EQ[illegible]

1. [illegible] EQ [illegible]
2. [illegible] EQ[illegible]
3. [illegible]

[illegible]

[illegible] Port [illegible] EQ [illegible] Port [illegible] EQ[illegible]

[illegible] 8 GT/s[illegible]Link Status 2 Regiser [illegible] Link Equanlization Requset 8 GT/s [illegible] 16 GT/s[illegible]16 GT/s Status Register [illegible] Link Equanlization Requset 16 GT/s [illegible] 32 GT/s[illegible]32 GT/s Status Register [illegible] Link Equanlization Requset 32 GT/s [illegible]

EQ [illegible]

1. [illegible] gen1[illegible]DSP [illegible] gen3 [illegible] preset [illegible]
2. EQ Phase 0[illegible]EC=0[illegible] 8 GT/s[illegible]USP [illegible] Preset [illegible] Coefficients [illegible] DSP[illegible] **DSP** [illegible] **Phase 0** [illegible]
3. EQ Phase 1[illegible]EC=1[illegible]DSP [illegible] USP [illegible] LF (Low Frequency[illegible]symbol7)[illegible]FS (Full Swing[illegible]symbol8) [illegible]Post-cursor (symbol9) [illegible] Phase 2/3 [illegible] 32 TS1 [illegible] EIEOS[illegible]
4. EQ Phase 2[illegible]EC=2[illegible]USP [illegible] Master[illegible] DSP [illegible] Tx Preset [illegible] USP [illegible] USP [illegible] Preset [illegible] coefficient [illegible]USP [illegible] 32 TS1 [illegible] EIEOS[illegible]DSP [illegible] USP [illegible] 65536 TS1 [illegible] EIEOS[illegible]
5. EQ Phase 3[illegible]EC=3[illegible]DSP [illegible] Master[illegible] USP [illegible] Tx Preset [illegible] DSP [illegible] DSP [illegible] Preset [illegible] coefficient [illegible]DSP [illegible] 32 TS1 [illegible] EIEOS[illegible]USP [illegible] DSP [illegible] 65536 TS1 [illegible] EIEOS[illegible]
6. EQ [illegible]EC=0[illegible]LTSSM [illegible] Recovery.RcvrLock -> Recovery.RcvrCfg -> Recovery.Idle -> L0[illegible] L0 [illegible] 32 TS1/TS2 [illegible] EIEOS[illegible]

image-20240726133733511

[illegible] PCIe EQ [illegible] DSP [illegible] phase1[illegible]

[illegible]

[illegible] phase0[illegible]

[illegible]

https://mangopapa.blog.csdn.net/article/details/124539607

ARM N2 [illegible]

ARM N2 [illegible]ARM Neoverse N2 reference design[illegible] ARM [illegible]Neoverse N2 [illegible] ARM [illegible] CPU [illegible]

1. [illegible]N2 [illegible]HPC[illegible]
2. [illegible]Neoverse N2 [illegible]
3. [illegible]ARM N2 [illegible]
4. [illegible]N2 [illegible]
5. [illegible]Neoverse N2 [illegible] ARM [illegible] ARM [illegible]

[illegible]ARM N2 [illegible]

ARM Neoverse N2 [illegible] ARM [illegible]IP[illegible] IP [illegible]

1. **ARM Neoverse N2 CPU** [illegible] N2 [illegible]N2 [illegible] ARMv9 [illegible]
2. **System Level Cache (SLC)**[illegible]ARM N2 [illegible]SLC[illegible]
3. **CCIX** [illegible] **CXL** [illegible]N2 [illegible] CCIX[illegible]Cache Coherent Interconnect for Accelerators[illegible] CXL[illegible]Compute Express Link[illegible]
4. **AMBA (Advanced Microcontroller Bus Architecture)** [illegible]AMBA [illegible]SoC[illegible] IP [illegible]N2 [illegible] AMBA [illegible]
5. **TrustZone** [illegible]ARM TrustZone [illegible]
6. **Mali GPU**[illegible] N2 [illegible] CPU [illegible] ARM [illegible] Mali GPU[illegible]
7. **SMMU (System Memory Management Unit)**[illegible]

[illegible] IP [illegible]ARM Neoverse N2 [illegible]

[illegible] High Level [illegible]

[illegible] Arm Neoverse N2 reference design Technical Overview

image-20240704104921533

[illegible] CMN-700[illegible] PICe [illegible] SMMU [illegible] NI-700[illegible] RNI [illegible] SCP[illegible] GIC [illegible] SNF [illegible] memory controller[illegible] core[illegible]

01 _PCIe [illegible].md

image-20240704105750294

image-20240704110348168

[illegible] 4 [illegible] S [illegible] CXL [illegible] CXL [illegible] PCIe[illegible] x16[illegible] x4 [illegible] 4 [illegible] controller [illegible] bifircation[illegible]

[illegible] CMN RNI[illegible]HNP [illegible] CCG [illegible] NI-700[illegible]NI-700 [illegible] controller [illegible] X16 [illegible] CCG [illegible] CXL [illegible] CCIX [illegible]

SMMU-700 [illegible] MMU[illegible] TBU[illegible]TCU [illegible] DTIIC [illegible]

MSI-IC [illegible] MSI [illegible] switch[illegible] controller [illegible] MSI [illegible] MSI [illegible]

ITS [illegible] PCIe [illegible] MSI [illegible] GIC[illegible] ITS [illegible] GIC [illegible]

int process [illegible]

NIC-450 [illegible] NI-700 [illegible]NIC-450 [illegible] core [illegible] cfg [illegible] controller [illegible]NI-700 [illegible] PCIe [illegible] AMBA [illegible] ACE [illegible]

PCIe [illegible]

Resource Degration

[toc]

[illegible] BAR [illegible] 5 [illegible] resource pool

image-20230907171728542

[illegible]

```
    InitializeResourcePool (&IoPool, PciBarTypeIo16);
    InitializeResourcePool (&Mem32Pool, PciBarTypeMem32);
    InitializeResourcePool (&PMem32Pool, PciBarTypePMem32);
    InitializeResourcePool (&Mem64Pool, PciBarTypeMem64);
    InitializeResourcePool (&PMem64Pool, PciBarTypePMem64);
```

[illegible] RC [illegible] prefetchable.

[illegible]

image-20230907112441286

[illegible] BAR [illegible] 11 [illegible]Prefetchable vs. Non-Prefetchable Memory Space[illegible]

- [illegible]Reads do not have side effects[illegible]
- [illegible]Write merging is allowed[illegible]

PCIe Technology 3.0 Chapter 4.1.2.2

1. DegradeResource() [illegible]

If any child device has both option ROM and 64-bit BAR, degrade its PMEM64/MEM64

requests in case that if a legacy option ROM image can not access 64-bit resources.

[illegible] PCD [illegible] `PcdPciDegradeResourceForOptionRom`

- X64 —- TRUE
- X32 ARM AARCH64 —- FALSE

[illegible] root bridge [illegible] bar0 [illegible] mem32[illegible] bar0[illegible] bar [illegible] 64bit[illegible]

1.2 Degrade Resource For Others

```
  //
  // Degrade resource if necessary
  //
  DegradeResource (Bridge, Mem32Node, PMem32Node, Mem64Node, PMem64Node);
```

⌧⌧⌧⌧ `BridgeSupportResourceDecode()`⌧⌧ bridge ⌧⌧⌧⌧⌧

⌧⌧⌧⌧ `MergeResourceTree()`⌧⌧⌧⌧ resoure tree

image-20230907153348997

⌧⌧⌧⌧⌧⌧⌧⌧⌧ PMem64 ⌧ PMem32⌧⌧⌧ bridge ⌧⌧⌧⌧⌧⌧⌧⌧⌧⌧⌧⌧⌧

⌧⌧⌧⌧⌧⌧⌧⌧NP-Mem64 ⌧⌧⌧⌧⌧⌧⌧⌧⌧ 32 ⌧⌧⌧⌧⌧⌧⌧ 32 ⌧⌧⌧⌧⌧⌧⌧

image-20230907100002700

image-20230906184552816

2. ⌧⌧⌧⌧⌧⌧

image-20230907103257590

image-20230907111105623

bit0 - 0 ⌧⌧ memory ⌧⌧⌧

bit 2:1 ⌧⌧ memory ⌧⌧⌧ 00 32bit memory ⌧⌧⌧ 10- 64 bit memory ⌧⌧ B

bit3 - 0 ⌧⌧ Non-prefetchable

NP-Mem64

2.2 Degrade Resource ⌧⌧⌧⌧

⌧⌧ `BridgeSupportResourceDecode()`⌧⌧⌧⌧⌧⌧ PCI _IO _DEVICE.Decodes

image-20230906184420587

image-20230907100002700

⌧⌧ bridge ⌧⌧⌧⌧⌧

```
#define EFI_BRIDGE_IO32_DECODE_SUPPORTED      0x0001
#define EFI_BRIDGE_PMEM32_DECODE_SUPPORTED    0x0002
#define EFI_BRIDGE_PMEM64_DECODE_SUPPORTED    0x0004
#define EFI_BRIDGE_IO16_DECODE_SUPPORTED      0x0008
#define EFI_BRIDGE_PMEM_MEM_COMBINE_SUPPORTED 0x0010
#define EFI_BRIDGE_MEM64_DECODE_SUPPORTED     0x0020
#define EFI_BRIDGE_MEM32_DECODE_SUPPORTED     0x0040
```

```
struct _PCI_IO_DEVICE {
   …
  //
  // The resource decode the bridge supports
  //
  UINT32                                    Decodes;
  …
}
```

PCI _IO _DEVICE.Decodes ⌧⌧⌧ DetermineRootBridgeAttributes()⌧⌧⌧⌧⌧⌧⌧⌧⌧ protocol ⌧⌧⌧

```
PciResAlloc->GetAllocAttributes (
                           PciResAlloc,
                           RootBridgeHandle,
                           &Attributes
                           );
```

- [illegible] root bridge[illegible] RC [illegible] root bridge [illegible]
- [illegible] bridge[illegible] bridge [illegible]
- IO capability – ACPI spec 7.5.1.3.6

image-20230906181159100

- Mem capability – PCIe spec 7.5.1.3.9

The bottom 4 bits of both the Prefetchable Memory Base[illegible]0x24[illegible] and Prefetchable Memory Limit registers are read-only, contain the same value, and encode whether or not the bridge supports 64-bit addresses

- 0 - the bridge supports only 32 bit addresses.
- 1 - the bridge supports 64-bit addresses.

[illegible] bridge [illegible]

image-20230907145320882

Prefetchable Memory Base 0x002**1**

Prefetchable Memory Limit 0x003**1**

[illegible] 0x28 [illegible] 32 [illegible] PMEM32 [illegible] PMEM64 flag [illegible] PMEM32 [illegible] PMEM32 [illegible] MEM32[illegible] PMEM64[illegible]PMEM32 [illegible] MEM32[illegible]

[illegible] PCIe Spec [illegible] EDK2 [illegible] Root Bridge [illegible] bridge [illegible] MEM64 Flag[illegible] bridge [illegible] NP-MEM64[illegible] NP-MEM64 tree [illegible] merge [illegible] NP-MEM32 tree [illegible]

[illegible] RC [illegible]

```
InitializeResourcePool (&IoPool, PciBarTypeIo16);
InitializeResourcePool (&Mem32Pool, PciBarTypeMem32);
// InitializeResourcePool (&PMem32Pool, PciBarTypePMem32);
// InitializeResourcePool (&Mem64Pool, PciBarTypeMem64);
InitializeResourcePool (&PMem64Pool, PciBarTypePMem64);
```

2.3 RC [illegible]

RC [illegible] Root Bridge [illegible]

Mem[illegible]PMem =**Mmio32Base**

Io.Base = Mem Limit [illegible] IO _SIZE

PMemAbove4G = **MmioBase**

MemAbove4G [illegible]

```
{  // PMem
 MAX_UINT64,
 0,
 0
}
```

image-20230907132349864

image-20230907133927772

```
  if (RootComplex->Mmio32Base != 0) {
      RootBridge->Mem.Base = RootComplex->Mmio32Base;
      RootBridge->Mem.Limit = RootComplex->Mmio32Base + RootComplex->Mmio32Size - 1;
      RootBridge->PMem.Base = RootBridge->Mem.Base;
      RootBridge->PMem.Limit = RootBridge->Mem.Limit;
      RootBridge->Io.Base = RootComplex->Mmio32Base + RootComplex->Mmio32Size -
AC01_PCIE_IO_SIZE;
      RootBridge->Io.Limit = RootBridge->Mem.Limit;
    }

    if (RootComplex->MmioBase != 0) {
      RootBridge->PMemAbove4G.Base = RootComplex->MmioBase;
      RootBridge->PMemAbove4G.Limit = RootComplex->MmioBase + RootComplex->MmioSize - 1;
    }
if ((Attributes & EFI_PCI_HOST_BRIDGE_COMBINE_MEM_PMEM) != 0) {
    RootBridgeDev->Decodes |= EFI_BRIDGE_PMEM_MEM_COMBINE_SUPPORTED;
  }

  if ((Attributes & EFI_PCI_HOST_BRIDGE_MEM64_DECODE) != 0) {
    RootBridgeDev->Decodes |= EFI_BRIDGE_MEM64_DECODE_SUPPORTED;
    RootBridgeDev->Decodes |= EFI_BRIDGE_PMEM64_DECODE_SUPPORTED;
  }

  RootBridgeDev->Decodes |= EFI_BRIDGE_MEM32_DECODE_SUPPORTED;
  RootBridgeDev->Decodes |= EFI_BRIDGE_PMEM32_DECODE_SUPPORTED;
  RootBridgeDev->Decodes |= EFI_BRIDGE_IO16_DECODE_SUPPORTED;
```

⊠⊠⊠

NP-MEM64 ⊠⊠⊠ NP-MEM32⊠⊠⊠ bar ⊠⊠⊠⊠⊠⊠⊠NP-MEM32 ⊠⊠⊠⊠⊠⊠⊠⊠⊠

⊠⊠⊠⊠⊠⊠⊠Linux kernel 5.18.0 ⊠⊠⊠⊠⊠⊠⊠⊠EDK2 ⊠⊠⊠⊠

⊠⊠ PCIe Spec ⊠⊠⊠⊠⊠⊠⊠ NP-MEM64 ⊠⊠⊠⊠⊠

image-20230907180903853

PCIe 3.0&4.0 Spec

image-20230915092058697

EDK2 ⊠ optionRom ⊠⊠⊠

[toc]

1. OpRom ⊠⊠⊠⊠

⊠⊠⊠⊠ 0x30 ⊠⊠⊠⊠⊠⊠⊠Expansion ROM Base Address Register⊠⊠⊠ option rom ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ Bridge ⊠⊠⊠⊠⊠⊠⊠⊠ 0x38⊠

image-20230908091633958

⊠⊠⊠⊠⊠

image-20230908091708328

⊠ 21bit ⊠⊠⊠⊠⊠⊠⊠⊠⊠ 8bit ⊠⊠⊠⊠⊠⊠⊠⊠

1.2 OpRom ⊠⊠

⊠⊠⊠ Plug and Play BIOS Specification ⊠⊠⊠⊠⊠⊠

EFI takes advantage of both the **PCI Firmware Specification** and the PE/COFF Specification to store EFI images in a PCI Option ROM. There are several rules that must be followed when constructing a PCI Option ROM:

From⊠ UEFI Spec 14.4.21 PCI Option ROMs

⊠⊠⊠⊠⊠⊠⊠ *PCI Firmware Specification 3.0* ⊠⊠⊠

2. ⊠⊠ OpRomImage

⊠⊠ `PciSearchDevice() --> GetOpRomInfo()` ⊠⊠⊠ OpRom ⊠⊠⊠⊠⊠ 1⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ RomSize⊠⊠⊠⊠⊠⊠⊠⊠⊠ PCI _IO _DEVICE ⊠⊠⊠⊠⊠⊠ 0 ⊠⊠⊠⊠⊠⊠ 1⊠⊠⊠ NOT FOUND.

⊠⊠ OpRom ⊠⊠⊠

⊠⊠ root bridge ⊠⊠ BAR ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ BAR ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ `ProcessOptionRom()` , ⊠⊠⊠⊠⊠ root ⊠⊠⊠⊠⊠⊠⊠⊠⊠ ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ optionRom ⊠⊠⊠⊠⊠⊠ root ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ `LoadOpRomImage`⊠⊠⊠⊠⊠ pci ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

⊠ `LoadOpRomImage` ⊠⊠ Rom ⊠⊠⊠⊠⊠ PCIR ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ rom ⊠⊠⊠⊠⊠⊠⊠root bridge ⊠⊠⊠⊠⊠ root bridge ⊠⊠⊠⊠ option rom size⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ root bridge ⊠ bar 0 ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ root bridge ⊠ BAR0 ⊠⊠⊠⊠⊠⊠ OptionRom Addr ⊠⊠⊠⊠⊠⊠ option rom ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ shadow⊠

⊠⊠⊠⊠⊠⊠⊠⊠ shadow oprom⊠⊠ program bar⊠⊠⊠⊠⊠ oprom⊠⊠⊠⊠⊠ bar ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ oprom ⊠⊠⊠⊠⊠⊠⊠⊠ root bridge ⊠⊠⊠⊠ bar0⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ shadow oprom⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

OpRom ⊠⊠⊠ bar ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠ oprom⊠⊠⊠⊠⊠⊠⊠⊠ bar ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

☒☒☒☒ rombar ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ 0x55aa☒☒☒☒☒☒☒☒☒☒☒ 512 ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ RomSize ☒☒☒☒ 3.1 ☒☒☒☒☒☒☒☒☒☒☒☒☒

☒☒☒☒☒ optionRom☒☒ FirstCheck ☒☒☒☒☒ FALSE☒☒☒ Pcir ☒☒☒☒☒☒ 0☒☒☒☒☒ 4 ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

☒☒ PCIr ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒image ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ legacy ☒☒☒☒☒ LegacyImageLength ☒☒☒

Legacy ☒☒☒☒☒☒Rom ☒ size ☒☒☒☒☒☒☒☒☒☒☒ rom ☒☒☒☒☒☒ 0 ☒☒☒ rom ☒☒☒☒☒☒☒

☒☒☒☒☒ rom ☒☒☒☒☒☒☒☒☒☒☒☒ PciDevice->PciIo ☒☒☒☒ optionRom ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

☒☒ OpRom Image

☒☒☒☒☒☒☒☒☒☒☒☒☒☒ `StartPciDevices` ☒☒☒☒☒☒ image ☒☒☒☒☒

image-20230908135043724

edk2MdeModulePkgBusPciPciBusDxePciOptionRomSupport.c

Resources

PCIe 5.0 Spec Chapter 7.5.1.2.4

https://blog.csdn.net/robinsongsog/article/details/51785335

https://zhuanlan.zhihu.com/p/343464819

https://www.cnblogs.com/free-1122/p/16611254.html

PCIe ☒☒☒☒

PCIe AER

PCIe Interrupt

PCIe Hot-Plug

PCIe Power Management

PCIe _DPC.md

[illegible]

DPC [illegible] downstream port containment[illegible] PCIe Spec 3.1 [illegible] root port [illegible] pcie switch [illegible] halt [illegible]DPC [illegible]

DPC [illegible] unmasked uncorrectable error [illegible] ERR _FATAL [illegible] ERR _NONFATAL [illegible] DPC [illegible]

[illegible] DPC [illegible]DP [illegible] DPC Trigger Status bit and DPC Trigger Reason field[illegible] LTSSM [illegible] disable [illegible] link[illegible]LTSSM [illegible] disable [illegible] DPC Trigger Status bit [illegible]

[illegible] DPC [illegible]LTSSM [illegible] detect [illegible]

[illegible] DPC [illegible] DPC [illegible] ERR _COR message[illegible]

image-20240704131847187

DPC Extended Capability structure

image-20240704131926844

[illegible] DPC extended capability [illegible] ID [illegible] 1Dh[illegible]

[illegible] RP PIO [illegible]

[illegible]

DPC [illegible]

DPC [illegible] DPC control register [illegible]

image-20240704132016601

[illegible] DPC Trigger Enable [illegible] DPC [illegible]

image-20240704132045698

- [illegible] DPC Trigger Enable field [illegible] 01b[illegible]DPC [illegible] ERR _FATAL Message [illegible] DPC[illegible]
- [illegible] DPC Trigger Enable field [illegible] 10b[illegible]DPC [illegible] ERR _NONFATAL or ERR _FATAL Message [illegible] DPC

[illegible] DPC

DPC [illegible] DPC [illegible]

- DPC [illegible] enable [illegible]
- [illegible] Port [illegible] DPC [illegible]
- DPC Software Triggering Supported bit in DPC Capability register [illegible] 0x1[illegible]

image-20240704132111525

☒☒☒☒☒☒☒☒☒☒ DPC☒

☒ 1b ☒ DPC Software Trigger bit in the DPC Control Register☒☒☒☒ Port ☒☒☒ DPC ☒☒☒

image-20240704132124832

☒☒☒☒ DPC Status Register ☒☒☒☒☒☒☒☒☒☒☒☒☒

image-20240704132148504

☒☒☒☒☒☒

DPC Trigger Status ☒ 1☒☒☒☒☒☒ DPC ☒☒☒

DPC Trigger Reason ☒ 0x11☒☒☒☒☒☒☒☒☒ DPC Trigger Reason Extension ☒☒☒

DPC Trigger Reason Extension ☒ 0x01☒☒☒☒☒☒ DPC software Trigger ☒☒

DPC ☒☒

☒☒ DPC ☒ DSP ☒☒☒☒ DPC ☒☒☒☒☒

DPC ☒☒☒☒ DPC Control Register ☒ DPC Interrupt Enable bit ☒☒☒

image-20240704132231738

DPC ☒☒☒☒☒☒ DPC Status Register ☒ DPC Interrupt Status bit ☒☒☒

image-20240704132253717

image-20240704132308721

DPC ERR _COR signaling

☒☒ DPC ☒ DSP ☒☒☒☒ ERR _COR ☒☒☒☒☒☒☒☒ Advanced Error Reporting (AER)☒

DPC ERR _COR ☒☒☒☒☒ DPC ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

DPC ERR _COR signaling is enabled by the DPC ERR _COR Enable bit in the DPC Control Register.

image-20240704132336098

☒☒☒☒☒

- the Correctable Error Reporting Enable bit in the Device Control Register
- **or** the DPC SIG _SFW Enable bit in the DPC Control Register is Set

PCIe Spec ☒☒ OS ☒☒ DPC ☒☒☒FW ☒☒ ERR _COR signaling

☒☒ DPC

☒☒☒☒☒☒☒ DPC Trigger Status bit in DPC Status Register ☒☒☒ Root port ☒☒ DPC☒

☒☒☒ LTSSM ☒☒☒☒☒☒☒ disbale ☒☒☒☒☒☒☒☒ Data Link Layer Link Active bit in the Link Status Register reads 0b ☒☒☒☒☒ DPC☒

image-20240704132410187

[illegible] Root Port [illegible]RP [illegible] DPC [illegible] DPC RP Busy bit reads 0b.

image-20240704132418536

DL _Active ERR _COR signaling

[illegible] DPC [illegible] active [illegible] DL _Active ERR _COR [illegible]

DL _Active [illegible] the Data Link Layer Link Active bit in the Link Status Register [illegible] DPC [illegible] bit[illegible]

[illegible] ERR _COR [illegible] ERR _COR [illegible] active [illegible]

[illegible] DL _Active ERR _COR signaling [illegible] DPC control [illegible]:

image-20240704132518904

[illegible] DPC ERR _COR [illegible]

- the Correctable Error Reporting Enable bit in the Device Control Register
- **or** the DPC SIG _SFW Enable bit in the DPC Control Register is Set

[illegible] DL _ACTIVE [illegible] ERR _COR [illegible] MSI/MSI-X [illegible] MSI/MSI-X [illegible] ERR _COR [illegible]

[illegible] INTx [illegible] DL _ACTIVE [illegible]INTx [illegible] ERR _COR [illegible]

eDPC

eDPC [illegible] DPC [illegible] PCIe Spec 3.1 [illegible] DPC [illegible] DPC [illegible] RP PIO[illegible]Root Port Programmed I/O[illegible]

RP PIO [illegible] PCIe [illegible] RP PIO [illegible]

[illegible] DPC capability [illegible] bit5 [illegible] RP [illegible] eDPC [illegible]

image-20240704132607043

PCIe [illegible]

PIO [illegible] PCIe [illegible] DMA [illegible] P2P[illegible]

image-20240704132910411

image-20240704132921038

image-20240704132935032

RP PIO [illegible]

[illegible]

Completion with Unsupported Request status (UR Cpl)

Completion with Completer Abort status (CA Cpl)

Completion Timeout (CTO) errors

[illegible]Configuration Requests[illegible] I/O Requests[illegible] Memory Requests[illegible] [illegible] RP PIO [illegible] 9 [illegible]

image-20240704133039272

AER [illegible] UR CA [illegible] CTO [illegible] RP PIO error control [illegible]

[illegible] RP [illegible] Completer [illegible] UR [illegible] CA error [illegible] AER [illegible]

[illegible] RP [illegible] Requester [illegible] UR [illegible] CA error [illegible] RP PIO [illegible] CTO error [illegible] AER [illegible] RP PIO [illegible]

[illegible] RP PIO [illegible] CTO error[illegible] AER [illegible] CTO error[illegible]

image-20240704133101020

image-20240704133108266

[illegible]

image-20240704133134383

RP PIO error control [illegible] DPC extended capability [illegible] RP PIO [illegible] AER capability [illegible]

The RP PIO Status, Mask, and Severity registers [illegible] AER [illegible] the Uncorrectable Error Status, Mask, and Severity registers[illegible]

[illegible] RP PIO [illegible]RP PIO Status Register [illegible] RP PIO log registers[illegible] RP PIO [illegible]

[illegible] RP PIO [illegible] RP PIO Status Register [illegible] DPC[illegible] RP PIO log registers [illegible]

[illegible] RP PIO [illegible] uncorrectable or advisory[illegible] RP PIO Severity Register [illegible]

[illegible] Severity bit [illegible] uncorrectable[illegible] DPC([illegible] DPC [illegible])[illegible] DPC [illegible]/[illegible] ERR _COR(if enabled)[illegible]

[illegible] Severity bit [illegible] Clear[illegible]([illegible] DPC)[illegible] ERR _COR(if enabled)[illegible]

The RP PIO Header Log Register, RP PIO ImpSpec Log Register, and RP PIO TLP Prefix Log Registers [illegible] RP PIO log registers[illegible]

The RP PIO First Error Pointer, RP PIO Header Log, and RP PIO TLP Prefix Log [illegible] AER [illegible] the First Error Pointer, Header Log, and TLP Prefix Log[illegible]

RP PIO Header Log [illegible] RP PIO error [illegible] TLP header;

RP PIO ImpSpec Log Register [illegible] request TLP [illegible] DPC capability [illegible] RP Log Size field [illegible] 5[illegible]

RP PIO TLP Prefix Log Register [illegible] RP PIO error [illegible] TLP Prefix[illegible] DW[illegible] DPC capability [illegible] RP Log Size field [illegible]

- Number = RP Log Size - 5[illegible] if RP Log Size <=9;
- Number =4[illegible]if RP Log Size >9;

Linux Kernel [illegible] DPC [illegible]

[illegible] Linux kernel 5.10.201 [illegible] DPC RAS [illegible] C2000 [illegible] DPC RAS [illegible]

image-20240704133341302

[illegible] dpc _probe [illegible] DPC [illegible]

image-20240704133356801

- line356[illegible] PCIe AER[illegible] DPC [illegible] -ENOTSUPP[illegible]
- line359[illegible] devm _request _threaded _irq [illegible] dev->irq [illegible] status [illegible]

[illegible]devm _request _threaded _irq [illegible] Linux [illegible]IRQ[illegible] dpc _irq [illegible] dpc _handler[illegible] top half [illegible] bottom half[illegible]

- line362[illegible]
- line368[illegible] DPC capability [illegible] control [illegible]
- line371[illegible] control [illegible] enable [illegible] Fatal error [illegible] control [illegible]

[illegible]

dpc _irq [illegible]

image-20240704133458974

- [illegible] DPC status [illegible] status [illegible] 1[illegible]
- [illegible] status [illegible] 0x8[illegible] dpc [illegible] 1b [illegible]
- [illegible] DPC [illegible] return IRQ _WAKE _THREAD;

[illegible] IRQ _WAKE _THREAD[illegible]

- [illegible] IRQ _HANDLED[illegible]

[illegible]

dpc _handler [illegible]

image-20240704133532559

- [illegible] dpc _process _error [illegible] pdev [illegible] DPC [illegible]
- [illegible] pcie _do _recovery [illegible]pdev[illegible]pci _channel _io _frozen [illegible] dpc _reset _link[illegible] DPC [illegible]
- [illegible] IRQ _HANDLED[illegible]

[illegible]

image-20240704133555716

- [illegible] status [illegible] source ID [illegible]
- [illegible] raw data[illegible]
- line287[illegible] DPC RP [illegible] reason [illegible] 3[illegible]ext _reason [illegible] 0[illegible] DPC [illegible] RP PIO [illegible] dpc _process _rp _pio _error[illegible]
- line289[illegible] reason [illegible] 0 [illegible]unmasked uncorrectable error[illegible] AER[illegible]Advanced Error Reporting[illegible] uce [illegible] AER [illegible]
- dpc _process _rp _pio _error [illegible]

[illegible]

pcie _do _recovery(pdev, pci _channel _io _frozen, dpc _reset _link);

[illegible] err.c [illegible] PCIe [illegible]

[illegible]

○ [illegible]

○ [illegible] I/O [illegible]

○ [illegible]/[illegible] I/O [illegible]/[illegible]

○ [illegible]“[illegible]”[illegible]

[illegible] **dpc _reset _link** [illegible] dpc.c [illegible] DPC [illegible]

[illegible] reset _subordinates[illegible]

- [illegible] PCIe [illegible]pcie _do _recovery [illegible]
- [illegible] Root Port[illegible]Downstream Port [illegible] RCEC[illegible]
- [illegible] Endpoint [illegible] Port [illegible]
- [illegible]reset _subordinates [illegible]

dpc _reset _link [illegible]

image-20240704133630816

line151[illegible] DPC [illegible]

line163[illegible] DPC [illegible]pcie _wait _for _link [illegible] inactive [illegible] 100 [illegible] true[illegible] false[illegible]

[illegible]

[illegible]

line172[illegible] status [illegible] bit0 [illegible] 1b[illegible] DPC [illegible] **PCIe 5.0 Spec [illegible]spec [illegible] disabled [illegible] DPC[illegible] 100ms[illegible] warning [illegible] DPC [illegible]** [illegible]

image-20240704133648074

line175[illegible] PCI _DPC _RECOVERED [illegible] PCI _ERS _RESULT _DISCONNECT[illegible]

line179[illegible] PCI _DPC _RECOVERED [illegible] PCI _ERS _RESULT _RECOVERED [illegible]

line184[illegible] DPC [illegible]

CXL

CXL [illegible]

[toc]

Overview

CXL [illegible] PCI Express 5.0 [illegible]

CXL [illegible] I/O(CXL.io[illegible] PCIe)[illegible](CXL.cache) [illegible](CXL.memory)[illegible]

CXL [illegible] I/O(CXL.io[illegible] PCIe)[illegible](CXL.cache) [illegible](CXL.memory)[illegible]

[illegible] CXL [illegible] PCIe [illegible] CXL [illegible] PCIe [illegible]CXL [illegible] PCIe Gen1 [illegible] Link Training[illegible] PCIe 5.0 [illegible] Alternate Negotiation Protocol [illegible] CXL [illegible] CXL[illegible] PCIe 5.0 [illegible] CXL[illegible] PHY[illegible]

image-20240725134114147

CXL Device Type

CXL [illegible] Device [illegible] CXL.io[illegible] CXL.cache [illegible] CXL.mem [illegible] CXL.io [illegible] CXL.cache [illegible] CXL.io [illegible] CXL.mem [illegible]

CXL.io [illegible] PCIe [illegible]I/O [illegible](DMA)[illegible]-[illegible] PCIe [illegible] CXL.cache [illegible] CXL.mem [illegible]

CXL.cache [illegible] Snoop Messages [illegible]

CXL.mem [illegible] CXL Memroy Device[illegible]CXL.mem [illegible] Load [illegible] Store [illegible]

image-20240725134229982

CXL Spec [illegible] CXL Device [illegible] CXL Device [illegible]

Type 1 CXL Device:[illegible] CXL.io [illegible] CXL.cache [illegible] CXL Device[illegible]

Type 2 CXL Device:[illegible] CXL Device[illegible] HBM [illegible] GPU [illegible]

Type 3 CXL Device:[illegible] CXL.io [illegible] CXL.mem [illegible] CXL Device[illegible] CXL Memory Expander[illegible]

image-20240725134250257

CXL.io Overview

CXL.io [illegible] I/O [illegible] non-coherent load/store [illegible] CXL.io [illegible] Flex Bus [illegible] PCIe [illegible] PCI Express 5.0 Base Specification[illegible]

image-20240725134501475

CXL.cache Overview

CXL.cache [illegible] [illegible]CXL.cache Channels[illegible] D2H [illegible] H2D[illegible]

CXL.cache [illegible] [illegible]CXL.cache Channels[illegible] D2H [illegible] H2D[illegible]

image-20240725135431891

H2D Request [illegible] snoops[illegible] snoops [illegible]H2D Response [illegible] ordering messages [illegible]CXL.mem Overview

CXL.mem Overview

CXL Memory Protocol [illegible] CXL.mem[illegible] CPU [illegible] Device Memory [illegible] Die [illegible] Compute Express Link(CXL)[illegible] Phy [illegible] Link Layer[illegible] Memory Controller [illegible] Host CPU [illegible] Memory Controller [illegible] Accelerator [illegible] Memory Controller [illegible]Volatile[illegible] Persistent[illegible]Flat[illegible] Hierarchical[illegible]

CPU Coherency Engine [illegible] CXL.mem [illegible] Request [illegible] Response [illegible]CPU Coherency Engine [illegible] CXL.mem Master[illegible] Mem Device [illegible] CXL.mem Subordinate[illegible]CXL.mem Master [illegible] CXL.mem [illegible] CXL.mem Subordinate [illegible] CXL.mem [illegible]

[illegible] Subordinate Device [illegible] Accelerator [illegible]CXL.mem Protocol [illegible] Device Coherency Engine(DCOH)[illegible]DCOH [illegible] CXL.mem [illegible]

CXL.mem [illegible] Master [illegible] Subordinate [illegible] “M2S"[illegible] Subordinate [illegible] Master [illegible] “S2M"[illegible]

[illegible] M2S [illegible]

- [illegible]–[illegible]Req[illegible]
- [illegible]–[illegible]RwD[illegible]

[illegible] S2M [illegible]

- [illegible] – [illegible] (NDR)
- [illegible]–[illegible]DRS[illegible]

[illegible] CXL [illegible]

image-20240725134549647

☒☒☒☒

Firmware ☒☒☒☒☒ Device ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒

- ☒☒ CXL2.0 Device☒☒☒☒☒ Component Register ☒ HDM Decoder ☒☒☒☒☒☒☒☒ Component Register ☒☒ CXL HDM Decoder Global Control Register (Offset 04h) ->HDM Decoder Enable ☒☒☒ DVSEC ID 0 ☒ Range Size ☒ Range Base ☒ Memory Address ☒☒☒☒
- ☒☒ CXL 1.1 Device☒☒☒☒☒ DVSEC ID 0 ☒ Range Size ☒ Range Base ☒☒☒☒
- ☒☒☒ CXL Root Port ☒ CXL Switch Upstream Port ☒ HDM Decoder ☒☒☒☒
- ☒☒ PCIe DVSEC for CXL Device ☒ CXL 2.0 Extensions DVSEC for Ports ☒☒☒ Mem _Enable☒

CXL HDM Decoder ☒☒☒

- ☒☒☒☒☒HDM Decoder ☒☒☒☒☒☒☒☒☒☒☒☒Host☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒HDM Decoder ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒
- ☒☒☒☒☒HDM Decoder ☒☒
- ☒☒☒☒☒☒☒☒☒HDM Decoder ☒☒☒☒☒☒☒☒☒☒☒☒☒ DRAM☒NVRAM ☒☒☒☒☒☒☒☒☒☒☒☒

CXL □□□

□□□□□□□□□□□□□□□CXL □□□□□□□□□□□□□□□□

□□□□ PCIe □□□□ □Configuration Space□□□□□□□□□□□□□□□ □□□□ MMIO □□ □Memory Mapped Space□□□□ □ Memory □□□□□□□□□

image-20240718140347127

DVSEC

□ PCIe □□□□□□□ DVSEC □□□□□□ CXL □□□□□□□□□□□□□□□□□□□□□□ CXL □□□□□□□ CXL □□□□□ vendor ID 0x1E98.

□□ CXL 3.0 □□□□□□ 9 □ CXL □□□ DVSEC□

- PCIe DVSEC for CXL Devices □□□ CXL PCIe DVSEC□□□ RCD□LD□SLD□FMLD □ CXL Device □ D0F0 □□□□□ DVSEC□□□□□ Device □□□□□□□□□□□ DVSEC □□□ Device □ PCIe Device □□ CXL Device □
- Non-CXL Function Map DVSEC □□□□□□ Device□□ Function □□□□□□ CXL.cachemem □ Device □ Function□
- CXL Extensions DVSEC for Ports □□□□ RP□DSP □ USP□□ PCIe □□□□□□□□□□ RCH-RCD □□□
- GPF DVSEC for CXL Ports □□□ CXL Port □□ GPF □□□□□□□□□□□ GPF Phase1□Phase2 □ Timeout □□
- GPF DVSEC for CXL Devices □□□ CXL Device □□ GPF □□□□□□□□□□□ GPF Phase2 □ Timeout □□ Phase2 □□□□□□
- PCIe DVSEC for Flex Bus Port □□□ Flex Bus Port DVSEC□□□□□□□□ Flex Bus □□□□□□□ Multi-Device □□ □□□□□□□□CXL 1.1 □□□□ CXL Port/Device □□□□□□□□□CXL □□□□□□ Modified TS □□ Flex Bus □□□□□□□□ □□□□□□□□□□□□□□□□□□□ Modified TS Info □□□□□□□ DVSEC □□□□□□□□□□□□ RCRB □ RCH/RCD□□ DVSEC □□ RCRB □□
- Register Locator DVSEC □□□□□□□□□□□□□□□ CXL □□□□□ Entry□
- MLD DVSEC □□□□□ FM □□□ LD□□□ MLD □□ Capability□□□□□□□□ LD □□□LD-ID □□□□□□
- PCIe DVSEC for Test Capability □□□ CXL Compliance □□□□ CXL Compliance □□□□□□□ DVSEC □□□□□ □□□□ DVSEC □□□□□□□□□

9 □ CXL DVSEC □□□□ DVSEC ID □□□□

image-20240709135151711

image-20240725171532850

□□ DVSEC □□□□□

image-20240725173843171

image-20240726101740598

Reg in MMIO

[illegible] 9 [illegible] PCIe [illegible] CXL [illegible] DVSEC[illegible] CXL [illegible] PCIe [illegible] PCIe [illegible] 4KB [illegible] CXL Component Register [illegible] Size [illegible] CXL [illegible] DVSEC [illegible] Component Register [illegible] Memory Mapped [illegible]MMIO[illegible] Memory [illegible]

[illegible] CXL 3.0[illegible]CXL [illegible] 6 [illegible] MMIO [illegible]

- **RCH DP RCRB** [illegible] RCH [illegible] 4KB [illegible] MEMBAR0 [illegible] ACPI [illegible]
- **RCH DP Component Register** [illegible] CXL [illegible] ARB/MUX [illegible] 64KB[illegible] RCH DP RCRB [illegible] 64b MEMBAR0 [illegible] RCH [illegible]
- **RCD UP RCRB** [illegible] RCD [illegible] 4KB [illegible] RCH DP RCRB [illegible] 4KB [illegible] RCH DP [illegible] Memory [illegible] MEMBAR0[illegible]RCD UP RCRB [illegible] RCD [illegible] RCH DP [illegible] MRd[illegible]CXL 1.1 [illegible] MRd/MWr [illegible]
- **RCD UP Component Register** [illegible] CXL [illegible] ARB/MUX [illegible] 64KB[illegible] RCRB [illegible] RCD[illegible] RCD UP RCRB [illegible] 64b MEMBAR0 [illegible] RCD [illegible] RCH DP [illegible] Memory [illegible] MEMBAR0[illegible] RCRB [illegible] RCD[illegible] Register Locator DVSEC [illegible]
- **Component Register for ALL Other CXL Components** [illegible] CXL Component [illegible] Register Locator DVSEC [illegible]
- **CXL Host Bridge Component Register (CHBCR)**[illegible]CXL Host Bridge [illegible] RP [illegible] Memory Interleaving[illegible] ACPI CEDT [illegible]

[illegible] RCRB [illegible] Components Register [illegible]CHBCR [illegible] Register Locator DVSEC [illegible] Component Registers for All Other CXL Components [illegible]

image-20240709135610271

[illegible]

[illegible] PCIe [illegible] CXL [illegible]blog

https://blog.csdn.net/weixin _40357487/article/details/132553156

CXL _in _CMN

[toc]

CXS [illegible]

ARM RD _N2 [illegible]chip to chip [illegible] CCIX2.0[illegible]CMN-700 TRM [illegible] ARM [illegible] CML _SMP[illegible]S [illegible] CXL [illegible] CXS [illegible] CCIX2.0[illegible] CML _SMP [illegible] CCIX2.0 [illegible]CMN [illegible] CCG [illegible] chip [illegible] CXL [illegible]

Coherent Multichip Link (CML) device = CCG

A given multi-chip link can be used for: • SMP (CML _SMP) connection • CXL device attachment

A CML device (CCG) can be configured to be used for **CML _SMP** connection or **CXL device attachment**.

For SMP systems, CCG block is required to enable multi-chip SMP communications over a **CXS issue B interface**.

[illegible] CXL [illegible] **CXS issue A interface**

CCG [illegible]

image-20240708132240058

image-20240708132453025

CCG [illegible]

The **CML RA** node type has the following MPAM modes:

- SMP mode The CML RA passes the MPAM field on the USER field of the request. When snooped, the CML RA receives the MPAM field and passes it through the CHI SNP MPAM field.
- Non-SMP mode The CML RA drops the MPAM field that is received on the CHI request. The CML RA also does not receive MPAM field on CML snoops in this mode.
- CXSA mode The CML RA passes the MPAM field on the USER field of the request, even though CXSA is in non-SMP mode. You can use a configuration bit to enable passing of MPAM attributes when in CXSA mode.

The **CML HA** node type has the following MPAM modes:

- SMP mode The CML HA receives MPAM fields through the USER field of the request and passes them through the CHI MPAM field. On incoming CHI transactions, the CML HA passes CHI MPAM values through on the USER field.

- Non-SMP mode The CML HA drops the MPAM values that it receives on CHI snoop. The CML HA does not receive the MPAM attributes on CML requests.

[illegible] CXL [illegible] enable **CXSA** mode in RA[illegible]

LA [illegible] enable CXL[illegible]

CCG SAM

CMN CCG [illegible] CXL [illegible] CMN700 [illegible] type3 [illegible] CXL type3 [illegible] memory [illegible] CMN SAM [illegible] CCG [illegible]

CXL _in _SCP

SCP [illegible] CXL [illegible]

[illegible] SCP EDK2

CCG [illegible]

- SHA-1: bc14e780c120fc656ce905d6e219078a0006da88
- module/cmn700: configuring CCG for mapping Host address to CXL mem

A Memory region is reserved for CXL Memory. CXL.Mem (0x3fe _0000 _0000) comes under 4TB Chip-0 memory and the whole region is by default configured as SCG. Configured CXL.Mem region in HNF-SAM HTG and CCG SA node IDs for HTGs in following order -

HNF _SAM _CCG _SA _NODEID _REG HNF _SAM _HTG _CFG3 _MEMREGION HNF _SAM _HTG _CFG2 _MEMREGION HNF _SAM _HTG _CFG1 _MEMREGION

CXL Memory region is accessible as Normal memory with above configuration.

This patch maps Host address space to CXL device mem area through CCG node, based on the CXL device memory size, which is discovered by CXL module. CXL module invokes runtime CMN700 API for mapping the host address space and configuring CCG node.

This patch also adds a flag "cxl _mem" in CCG _Config structure for identifying host region and configuration reserved for CXL device memory purpose and thus differentiating from Remote chip memory.

Signed-off-by: Sayanta Pattanayak sayanta.pattanayak@arm.com Change-Id: I988f471db6a6a55f97320519413daedd8ea524fb

config _cmn700.c

N2 [illegible] 0-0x40000000000 [illegible] SCG [illegible]

[illegible] HNSAM [illegible] 0-0x3fe00000000 [illegible] CCG-cxl _mem [illegible]

```
#if (PLATFORM_VARIANT == 1)
static const struct mod_cmn700_ccg_config ccg_config_table_chip_0[] = {
    {
        .remote_rnf_count = 0,
        .remote_mmap_table = {
            .base = UINT64_C(0x3fe00000000),
            .size = UINT64_C(8) * FWK_GIB,
            .type = MOD_CMN700_REGION_TYPE_CCG,
        },
        .ra_mmap_table = {
            {
                .base = UINT64_C(0x3fe00000000),
                .size = UINT64_C(8) * FWK_GIB,
                .remote_haid = 0,
            },
            { 0 }
        },
        .remote_agentid_to_linkid_map = {
            {
                .remote_agentid_start = (RNF_PER_CHIP * CHIP_0),
                .remote_agentid_end = (RNF_PER_CHIP * CHIP_0)
            },
        },
        .smp_mode = false,
        .cxl_mem  = true,
    },
};
#endif
```

☒☒☒☒☒ entry☒☒☒ 8G☒☒☒☒☒☒☒☒ flag cxl _mem☒☒☒☒ CCG☒

```
static int cmn700_setup_rnsam_ccg_regions(void)
{
    const struct mod_cmn700_config *config;
    const struct mod_cmn700_mem_region_map *region;
    struct cmn700_rnsam_reg *rnsam;
    unsigned int count;
    unsigned int cxra_ldid;
    unsigned int cxra_node_id;
    unsigned int idx;
    uint32_t bit_pos;
    uint32_t group;

    config = ctx->config;
    /* Do configuration for CCG Nodes */
    for (idx = 0; idx < config->ccg_table_count; idx++) {
        region = &config->ccg_config_table[idx].remote_mmap_table;
        if (region->type != MOD_CMN700_REGION_TYPE_CCG) {
            return FWK_E_DATA;
        }

        FWK_LOG_INFO(
            MOD_NAME "  [0x%llx - 0x%llx] %s",
            region->base,
            region->base + region->size - 1,
            mmap_type_name[region->type]);

        /* If the region is for extended memory area like CXL.Mem
         * and connected through CCG then the region shouldn't be
         * marked as Non-Hash region. CXL.Mem region should be part
         * of Hashed cache group area.
         */
        if (config->ccg_config_table[idx].cxl_mem == true)
            continue;

        for (count = 0; count < ctx->internal_rnsam_count; count++) {
            rnsam = ctx->internal_rnsam_table[count];
```

[illegible] CCG [illegible] CXL.Mem[illegible] non-hash[illegible] hash[illegible]

```
static int map_ccg_for_cxl_mem(uint64_t size)
{
    uint32_t idx;
    const struct mod_cmn700_config *config = ctx->config;
    const struct mod_cmn700_ccg_config *ccg_config;

    cmn700_rnsam_stall();

    /* Do configuration of CCG Node for mapping remote CXL Mem area. */
    for (idx = 0; idx < config->ccg_table_count; idx++) {
        ccg_config = &(config->ccg_config_table[idx]);
        if (ccg_config->cxl_mem == true)
            ccg_setup_for_remote_mem(size, ctx, ccg_config);
    }

    cmn700_rnsam_unstall();

    return FWK_SUCCESS;
}
```

[illegible] API[illegible] CCG CXL.Mem

[illegible] rnsam stall [illegible] unstall [illegible] rnsam [illegible] default node[illegible]

[illegible] STATUS [illegible] 0x2[illegible] unstall[illegible]enable RNSAM[illegible]

[illegible] online [illegible] RNSAM[illegible] stall [illegible]

image-20240708100321441

image-20240708100309041

ccg _setup _for _remote _mem

```
int ccg_setup_for_remote_mem(
    uint64_t size,
    struct cmn700_device_ctx *ctx,
    const struct mod_cmn700_ccg_config *ccg_config)
{
    uint64_t reg_val;
    unsigned int index;
    unsigned int ccg_ldid;
    struct cmn700_hnf_reg *hnf_reg;
    struct cmn700_ccg_ra_reg *ccg_ra_reg;

    cmn700_ccg_ctx.is_prog_for_port_agg = false;
    /* Enable CXSA */
    ccg_ldid = get_ldid(ctx, cmn700_ccg_ctx.is_prog_for_port_agg);
    ccg_ra_reg = ctx->ccg_ra_reg_table[ccg_ldid].ccg_ra_reg;

    for (index = 0; index < ctx->hnf_count; index++) {
        /* Programming sequence to enable CXL.mem regions inside HNSAM */

        hnf_reg = (struct cmn700_hnf_reg *)ctx->hnf_node[index];

        /* Configuring CCG SA node IDs for HTGs in the HNSAM */
        reg_val = hnf_reg->HNF_SAM_CCG_SA_NODEID_REG[0];
        reg_val &= ~(CMN700_HNF_SAM_CCG_SA_NODEID_MASK);
        reg_val |= get_node_id(ccg_ra_reg);
        hnf_reg->HNF_SAM_CCG_SA_NODEID_REG[0] = reg_val;

        /* CXSA/CXLSA aggregated SA selection function */
        /* □□□□□□ region □□□□□ SN NODE□□ 8SN node， □□□ CXSA □□ */
        hnf_reg->HNF_SAM_HTG_CFG3_MEMREGION[0] |=
            (CMN700_HNF_SAM_HTG_MODE_CXSA <<
            CMN700_HNF_SAM_HTG_SN_MODE_POS);

        /* 64B interleaved */
        /* CXSA □□□□ interleaved □□□□□□□□□□□□□□□□□□□□□□ 0□64B */
        reg_val = hnf_reg->HNF_SAM_HTG_CFG3_MEMREGION[0];
        reg_val &= ~(CMN700_HNF_SAM_HTG_SA_DEVICE_INTERLEAVE_CNTL_MASK <<
               CMN700_HNF_SAM_HTG_SA_DEVICE_INTERLEAVE_CNTL_POS);
        hnf_reg->HNF_SAM_HTG_CFG3_MEMREGION[0] |= reg_val;

        /* 1 CXSA/CXLSA port used */
        /* □□□□□□□□□□□□□□□□□□ 0 */
        reg_val = hnf_reg->HNF_SAM_HTG_CFG3_MEMREGION[0];
        reg_val &= ~(CMN700_HNF_SAM_HTG_SA_PORTS_CNT_MASK <<
               CMN700_HNF_SAM_HTG_SA_PORTS_CNT_POS);
        hnf_reg->HNF_SAM_HTG_CFG3_MEMREGION[0] |= reg_val;

        /* htg_region_end_addr[51:20] = end address of HTG region */
        reg_val = ccg_config->ra_mmap_table[0].base +
              (size - 1);
        reg_val &= ~(CMN700_HNF_SAM_HTG_REGION_ADDR_RES_MASK);
        hnf_reg->HNF_SAM_HTG_CFG2_MEMREGION[0] |= reg_val;

        /* htg_region_base_addr[51:20] = start address of HTG region */
        reg_val = ccg_config->ra_mmap_table[0].base;
        reg_val &= ~(CMN700_HNF_SAM_HTG_REGION_ADDR_RES_MASK);
```

```
        hnf_reg->HNF_SAM_HTG_CFG1_MEMREGION[0] |= reg_val;

        /* Configuring HTG region as Valid */
        hnf_reg->HNF_SAM_HTG_CFG1_MEMREGION[0] |=
            (UINT64_C(0x1) << CMN700_HNF_SAM_HTG_REGION_VALID_POS);
    }

    /*
     * Program the CXRA SAM with the address range and the corresponding
     * remote HAID.
     */
    program_ccg_ra_sam_addr_region(ctx, ccg_config);

    update_cxl_mem_region(ccg_config->ra_mmap_table[0].base,
                          ccg_config->ra_mmap_table[0].size);

    return FWK_SUCCESS;
}
```

⊠⊠⊠⊠

⊠⊠⊠⊠⊠⊠⊠⊠cxl ⊠⊠⊠cxsa enable ⊠⊠

- RA ⊠ SAM region: ⊠⊠
- RA ⊠ SAM valid⊠valid ⊠⊠⊠

image-20240708143737820

image-20240708143928830

- CCLA ⊠ flex bus control⊠ MEM CACHE
- CCLA ⊠ config control⊠ MEM CACHE

image-20240708143808070

image-20240708144723101

image-20240708144143007

image-20240708144516623

image-20240708144538152

link ⊠

image-20240708145141381

image-20240708145159022

image-20240708145618083

image-20240708145629800

CXL ☒☒

[toc]

CXL ☒☒☒☒

☒☒☒☒☒☒☒☒☒☒ CXL ☒☒ PCIe☒☒☒☒☒☒☒☒

image-20240709112604509

image-20240709112653161

PCIe DVSEC for CXL Devices ☒☒☒ CXL PCIe DVSEC☒☒☒ RCD☒LD☒SLD☒FMLD ☒ CXL Device ☒ D0F0 ☒☒☒☒☒ DVSEC☒☒☒☒☒ Device ☒☒☒☒☒☒☒☒☒☒☒ DVSEC ☒☒☒ Device ☒ PCIe Device ☒☒ CXL Device ☒

CXL2.0

☒ PCIe ☒☒☒

CXL1.1

RCiEP

HDM Decoder

DDR

DDR Introduction

[toc]

Introduction

SDRAM vs. SRAM

- Synchronous Dynamic Random Access Memory
- Static RAM

image-20240725093325332

☒☒☒☒☒ SRAM☒☒☒☒ SDRAM☒

image-20240725103832027

DDR3 SDRAM ☒☒☒☒☒☒☒

image-20240725094537472

- ☒☒☒☒☒☒☒☒☒☒☒☒☒☒18b ☒☒☒
- ☒☒☒☒☒☒☒☒☒☒
- 8 ☒ bank -> BA[2:0], 3 ☒ bit ☒☒ 8 ☒☒☒☒
- ☒☒☒☒☒☒ A☒15b ☒☒☒
- ☒☒☒☒☒☒ 10b;
- 8192☒ ☒☒☒☒☒☒☒☒☒☒

☒☒☒☒**Row Address**☒☒16 ☒☒A0-A15☒

☒☒☒☒**Column Address**☒☒10 ☒☒A0-A9☒

Bank ☒☒☒Bank Address☒☒3 ☒☒BA0-BA2☒

4 ☒☒☒☒☒☒☒☒☒☒☒☒☒☒8 ☒☒☒

image-20240725101809984

256Mb Configuration

| ☒☒ | 256Mb x 4 | 128Mb x 8 | 64Mb x 16 | | ———————— | ———— | ——— | ——— | | # of Banks | 8 | 8 | 8 | | Bank Address | BA0 - BA2 | BA0 - BA2 | BA0 - BA2 | | Auto precharge | A10/AP | A10/AP | A10/AP | | BC switch on the fly | A12/BC# | A12/BC# | A12/BC# | | Row Address | A0 - A13 | A0 - A13 | A0 - A12 | | Column Address | A0 - A9,A11 | A0 - A9 | A0 - A9 | | Page size | 1 KB | 1 KB | 2 KB |

2Gb Configuration

| ☒☒ | 512Mb x 4 | 256Mb x 8 | 128Mb x 16 | | ———————— | ———— | ——— | ———- | | # of Banks | 8 | 8 | 8 | | Bank Address | BA0 - BA2 | BA0 - BA2 | BA0 - BA2 | | Auto precharge | A10/AP | A10/AP | A10/AP | | BC

switch on the fly | A12/BC# | A12/BC# | A12/BC# | | Row Address | A0 - A14 | A0 - A14 | A0 - A13 | | Column Address | A0 - A9,A11 | A0 - A9 | A0 - A9 | | Page size | 1 KB | 1 KB | 2 KB |

DDR4 SDRAM □□□□□□

image-20240725095148736

□□□□Row Address□□16 □□A0-A15□

□□□□Column Address□□10 □□A0-A9□

Bank □□□Bank Address□□2 □□BA0-BA1□

Bank □□□□Bank Group Address□□2 □□BG0-BG1□

DDR5 SDRAM □□□□□□

image-20240725100117145

□□□□Row Address□□16 □□A0-A15□

□□□□Column Address□□10 □□A0-A9□

Bank □□□□Bank Group Address□□2 □□BA0-BA1□

Bank □□□Bank Address□□3 □□BG0-BG2□

□□□□□Command Address□□□□□□ CA □□

□□□□

1. Channel□□□□□□□□□□□□□□□
2. Module□□□□□□□□□□□□□□□□□□□ Channel □
3. Rank□□□ DDR SDRAM □□□□□□□□□□□□□ Module □□□□□□□□□ Rank□□□ Rank □□□□□□□ Rank □□□□□□□□ □ CS _n□□□□□□□□□□□□ SDRAM □□
4. Chip□□□□□□ DDR SDRAM □□□□□□□□□□□□ 64 □□ Rank□□□□ 8 □ x8 □ Chip □□□□□□□□
5. Bank Group□DDR4 □□□□□ Bank Group □□ DDR4 □□□ Bank Group □□□□□□□ tCCD _L
6. Bank□□□ Bank □□□□□□ Row □□□□□□□□ Bank □□ Activate/Precharge □□□
7. Row□Activate/Precharge □□□
8. Column□□□ Column □□ n □ Cell□n □ SDRAM □□□
9. Cell□□□ Cell □□ 1 bit □□□

□□□□

- □□**Row**□□□□**Column**□□DDR □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- **Page**□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 1024 □□□□□□□□□□□□□□□□□□□□□□□□□□□ 1024 □□□

□□□□□□□□□□□□ 2K page □□□□□□□□ 1024 □□□□□□□ x16 □□□□□ 2 □ B□□□□ 2KB□

8N [illegible]

[illegible] 32 [illegible] 4 [illegible]8 [illegible] 8N[illegible]

[illegible] 32 [illegible]

[illegible]

[illegible]

[illegible]DQ[illegible]DQS [illegible] DM[illegible]

image-20240725110000495

fly-by [illegible] DRAM [illegible]

CLK/CLK#[illegible]DQS[illegible]AD [illegible] CMD [illegible]

DQ [illegible]

[illegible] DQ [illegible] 8 [illegible]kua [illegible]DQ [illegible]

Write Leveling

[illegible] Fly-by Topology [illegible]

[illegible] SDRAM [illegible] DQS [illegible] CK [illegible]DQ[illegible]DQS[illegible]

DQS[illegible]Data Strobe[illegible] CK[illegible]Clock[illegible]

DQS [illegible]

CK [illegible]

DQ [illegible] DQ [illegible]

[illegible] DQS [illegible] DQS-DQS# [illegible] CK [illegible] DQ [illegible] DDR3 [illegible] 01010101 [illegible] DQS-DQS# [illegible] DQ [illegible] 0 [illegible] 1 [illegible] Write leveling [illegible]

[illegible]

1. [illegible] SDRAM [illegible] Write Leveling [illegible] SDRAM [illegible] DQS [illegible] CK [illegible] DQ
2. [illegible] DQS [illegible] DQS [illegible] DQ [illegible] 0-1 [illegible]`001111111111111111110000`[illegible] 0[illegible] 1[illegible] 0
3. [illegible] DQS [illegible] DQ [illegible] 0 [illegible] 1 [illegible]DQS [illegible] CK [illegible]
4. [illegible] SDRAM [illegible] Write Leveling [illegible]

Read Leveling

[illegible] SDRAM [illegible]

[illegible]

[illegible]

1. [illegible] SDRAM [illegible] Pattern [illegible]
2. [illegible] 0 [illegible]
3. [illegible]
4. [illegible] 1[illegible] 0
5. [illegible] 1 [illegible]

[illegible] 0-1 [illegible]Write Leveling [illegible] 0-1 [illegible] 0-1 [illegible] Read Leveling [illegible] Read Centering[illegible]

Spec

SDRAM [illegible] JEDEC [illegible]

- JESD79F: DDR SDRAM
- JESD79-2F: DDR2 SDRAM
- JESD79-3F: DDR3 SDRAM
- [JESD79-4D: DDR4 SDRAM](https://www.jedec.org/document _search?search _api _views _fulltext=jesd79-4 ddr4)
- JESD79-5B: DDR5 SDRAM

[illegible] DDR [illegible] LPDDR [illegible]

- JESD209B: LPDDR SDRAM
- JESD209-2F: LPDDR2 SDRAM
- JESD209-3C: LPDDR3 SDRAM
- JESD209-4D: LPDDR4 SDRAM
- JESD209-5B: LPDDR5 SDRAM

[illegible] HBM [illegible] SDRAM [illegible]

- JESD235D: HBM
- JESD238A: HBM3

[illegible] GDDR SGRAM [illegible]

- SDRAM3.11.5.8 R16.01: GDDR4 SGRAM
- JESD212C.01: GDDR5 SGRAM
- JESD232A.01: GDDR5X SGRAM
- JESD250D: GDDR6 SGRAM

[illegible]

https://www.cnblogs.com/sky-heaven/p/15948268.html

Blog

https://zhuanlan.zhihu.com/p/26327347

LeetCode

⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

| ⊠⊠⊠⊠ | ⊠⊠⊠⊠⊠⊠⊠⊠⊠ | ⊠⊠⊠⊠⊠⊠⊠⊠⊠ | ⊠⊠⊠⊠⊠⊠⊠⊠⊠ | ⊠⊠⊠⊠⊠ | ⊠⊠⊠⊠⊠⊠ | | ————————- | ——————— | ——————— | ——————— | ———- | ———— | | ⊠⊠⊠⊠ (Bubble Sort) | O(n) | O(n^2) | O(n^2) | O(1) | ⊠⊠ | | ⊠⊠⊠⊠ (Selection Sort) | O(n^2) | O(n^2) | O(n^2) | O(1) | ⊠⊠⊠ | | ⊠⊠⊠⊠ (Insertion Sort) | O(n) | O(n^2) | O(n^2) | O(1) | ⊠⊠ | | ⊠⊠⊠⊠ (Quick Sort) | O(n log n) | O(n log n) | O(n^2) | O(log n) | ⊠⊠⊠ | | ⊠⊠⊠⊠ (Merge Sort) | O(n log n) | O(n log n) | O(n log n) | O(n) | ⊠⊠ | | ⊠⊠⊠ (Heap Sort) | O(n log n) | O(n log n) | O(n log n) | O(1) | ⊠⊠⊠ |

C ⊠⊠⊠⊠

[toc]

C ⊠⊠⊠⊠⊠⊠

static ⊠⊠

extern ⊠⊠

⊠⊠⊠⊠⊠⊠⊠⊠

C ⊠⊠⊠⊠⊠⊠

C ⊠⊠⊠⊠⊠⊠

C ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

⊠⊠ C ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠

image-20240729163935134

C ⊠⊠⊠⊠⊠⊠⊠

```
#include <stdio.h>

int check_endianness() {
    union {
        unsigned int i;
        unsigned char c[4];
    } test_union;

    test_union.i = 0x01020304;

    if (test_union.c[0] == 1) {
        return 1; // ⊠⊠
    } else {
        return 0; // [illegible]
    }
}

int main() {
    if (check_endianness()) {
        printf("[illegible] (Big-endian)\n");
    } else {
        printf("[illegible] (Little-endian)\n");
    }
    return 0;
}
```

test _union.i [illegible] 0x01020304[illegible] 32 [illegible] 4 [illegible]

[illegible]Big-endian[illegible]01 02 03 04[illegible]

[illegible]Little-endian[illegible]04 03 02 01[illegible]

[illegible] 64 [illegible] 0[illegible]1 [illegible]

Brian Kernighan [illegible] 1[illegible]

Brian Kernighan [illegible]

`x=x&(x-1)`

image-20240729164839182

```
#include <stdio.h>
#include <stdint.h>

void count_bits(uint64_t number, int *count_zeros, int *count_ones) {
    *count_ones = 0;
    while (number) {
        number &= (number - 1);
        (*count_ones)++;
    }
    *count_zeros = 64 - *count_ones;
}

int main() {
    uint64_t number = 0b11010101;  // [illegible]
    int count_zeros, count_ones;
    count_bits(number, &count_zeros, &count_ones);
    printf("Number of 0s: %d, Number of 1s: %d\n", count_zeros, count_ones);
    return 0;
}
```

MSB LSB

☒☒ 64 ☒☒☒☒☒☒☒☒☒☒☒☒☒Highest Significant Bit, MSB☒☒☒☒☒☒☒☒Least Significant Bit, LSB☒☒☒☒☒

```c
#include <stdio.h>
#include <stdint.h>

// □□□□□□□□LSB□□□□
int find_lsb(uint64_t number) {
    if (number == 0) return -1;  // □□□□ 0□□□□□□
    int position = 0;
    while ((number & 1) == 0) {
        number >>= 1;
        position++;
    }
    return position;
}

// □□□□□□□□MSB□□□□
int find_msb(uint64_t number) {
    if (number == 0) return -1;  // □□□□ 0□□□□□□
    int position = 63;
    while ((number & (1ULL << position)) == 0) {
        position--;
    }
    return position;
}

int main() {
    uint64_t number = 0b11010101;  // □□□
    int lsb_position = find_lsb(number);
    int msb_position = find_msb(number);

    if (lsb_position != -1) {
        printf("LSB position: %d\n", lsb_position);
    } else {
        printf("The number has no LSB (number is 0).\n");
    }

    if (msb_position != -1) {
        printf("MSB position: %d\n", msb_position);
    } else {
        printf("The number has no MSB (number is 0).\n");
    }

    return 0;
}
```

memory copy

```c
void *memcpy(void *dest, const void *src, size_t count)
{
        char *tmp = dest;
        const char *s = src;

        while (count--)
                *tmp++ = *s++;
        return dest;
}
```

Bit Map

[toc]

338. [illegible]

```
vector<int> ans;
    int countOnes(int nums) {
        int numOnes = 0;
        while(nums){
            nums &= (nums - 1);
            numOnes++;
        }
        return numOnes;
    }
    vector<int> countBits(int n) {
        for (int index = 0; index <= n; index++) {
            ans.push_back(countOnes(index));
        }
        return ans;

    }
```

Brian Kernighan [illegible] 1[illegible]

Brian Kernighan [illegible]

`x=x&(x-1)`

[illegible]

[toc]

[illegible]

[illegible]

while [illegible]

- [illegible] mid [illegible]
- [illegible]

```
++
class Solution {
public:
    int searchInsert(vector<int>& nums, int target) {
        int left = 0;
        int right = nums.size() - 1;
        while(left<=right) {
            // [illegible]left + right [illegible] int max
            // [illegible]
            int mid = left + ((right -left) >> 1);
            if (nums[mid] > target) {
                // [illegible]
                right = mid - 1;
            } else if (nums[mid] < target) {
                // [illegible]
                left = mid + 1;
            } else {
                return mid;
            }
        }
        //[illegible]
        return left;

    }
};
```

74. [illegible]

[illegible] for [illegible]

[illegible] mid [illegible]

```
++
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int left = 0;
        int right = matrix.size() * matrix[0].size() - 1;

        while (left <= right) {
            int mid = left + ((right - left) >> 1);
            int mid_value = matrix[mid/matrix[0].size()][mid%matrix[0].size()];
            // printf("%d \n", mid_value);

            if (target == mid_value) {
                return true;
            } else if (target > mid_value) {
                //right
                left = mid + 1;
            } else {
                right = mid - 1;
            }
        }
        // for (int x = 0; x < matrix.size(); x++) {
        //     for (int y = 0; y < matrix[0].size(); y++) {
        //         // printf("x-y: %d - %d [illegible] %d\n", x, y, matrix[x][y]);
        //         if (target == matrix[x][y]) {
        //             return true;
        //         }
        //     }
        // }
        return false;

    }
};
```

34. [illegible]

[illegible]

[illegible] first [illegible]last [illegible]

```
++
class Solution {
public:
    vector<int>  ans;
    vector<int> searchRange(vector<int>& nums, int target) {
        int left = 0;
        int right = nums.size() - 1;

        int first = -1;
        int last = -1;
```

```
// □□□□□□
if (nums.size() == 0) {
    ans.push_back(-1);
    ans.push_back(-1);
    return ans;
}

while (left<=right)
{
    /* code */
    int mid = left + ((right - left) >> 1);
    if (nums[mid] == target){
        first = mid;
        right = mid -1;
    } else if (nums[mid] > target) {
        //
        right = mid -1;
    } else {
        left = mid + 1;
    }
}

left = 0;
right = nums.size() - 1;

while (left<=right)
{
    /* code */
    int mid = left + ((right - left) >> 1);
    if (nums[mid] == target){
        last = mid;
        left = mid +1;
    } else if (nums[mid] > target) {
        //
        right = mid -1;
    } else {
        left = mid + 1;
    }
}

ans.push_back(first);
ans.push_back(last);
return ans;
```

```
    }
};
```

33. [illegible]

[illegible]

[illegible]

[illegible]

++

```
class Solution {
public:
    int search(vector<int>& nums, int target) {
        int left=0,right=nums.size()-1;
        while(left<=right){
            int mid=left+(right-left)/2;
            if(target==nums[mid]) return mid;
            // [illegible]
            if(nums[left]<=nums[mid]){
                // [illegible]
                if(target>=nums[left]&&target<nums[mid]){
                    right=mid-1;
                }else{
                    // [illegible]
                    left=mid+1;
                }
            }else{
                // [illegible]
                if(target>nums[mid]&&target<=nums[right]){
                    left=mid+1;
                }else{
                    right=mid-1;
                }
            }
        }
        return -1;
    }
};
```

153. [illegible]

[illegible]

[toc]

160. [illegible]

[illegible]

[illegible]

[illegible]

- [illegible]

++

```
class Solution {
public:
    ListNode *getIntersectionNode(ListNode *headA, ListNode *headB) {

        if (headA == NULL || headB == NULL) {
            return NULL;
        }
        ListNode *pa = headA;
        ListNode *pb = headB;
        while (pa != pb) {
            pa = (pa == NULL ? headB : pa->next);
            pb = (pb == NULL ? headA : pb->next);
        }
        return pa;
    }
};
```

206. [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

++

```
ListNode* reverseList(ListNode* head) {
        ListNode *pre = nullptr;
        ListNode *curr = head;
        while (curr □= NULL) {
            /* [illegible] temp [illegible] */
            ListNode * temp = curr->next;
            curr->next = pre;
            pre = curr;
            curr = temp;
        }

        return pre;
```

92. [illegible] II

[illegible]

234. [illegible]

[illegible] index [illegible]

```
++
vector<int> nums;

        ListNode* p = head;

        while (p != nullptr) {
            nums.push_back(p->val);
            p = p->next;
        }

        for (int i = 0, j = (int)nums.size() - 1; i < nums.size()/2; i ++, j--) {
            if (nums[i] != nums[j]) {
                return false;
            }
        }
        return true;
```

141. [illegible]

[illegible]

[illegible] false.

```
++
       if (head == NULL || head->next == NULL) {
            return false;
        }

        ListNode * fast = head->next;
        ListNode * slow = head;

        while(fast != slow) {
            if (fast == NULL || fast->next == NULL) {
                return false;
            }

            fast = fast->next->next;
            slow = slow->next;
        }
        return true;
```

[illegible]

[illegible]

```
class Solution {
public:
    bool hasCycle(ListNode *head) {
        unordered_set<ListNode*> table;
        while(head) {
            if (table.find(head) != table.end()) {
                return true;
            }
            table.insert(head);
            head = head->next;
        }
        return false;

    }
};
```

142. [illegible] II

[illegible]

[illegible]

[illegible]

[illegible]

[illegible] a[illegible] b

f=2s

f=s+nb[illegible]f [illegible] s [illegible] n [illegible]

[illegible]

s=nb

[illegible]

[illegible] Jb [illegible]

[illegible]

++

```
class Solution {
public:
    ListNode *detectCycle(ListNode *head) {
        if (head == NULL || head->next == NULL) {
            return NULL;
        }
        ListNode* fast = head;
        ListNode* slow = head;

        while(true) {
            if (fast == NULL || fast->next == NULL) {
                return NULL;
            }
            fast = fast->next->next;
            slow = slow->next;
            if (fast == slow) {
                break;
            }
        }

        if (fast == slow) {
            ListNode* ptr = head;
            while (ptr != slow) {
                slow = slow->next;
                ptr = ptr->next;
            }
            return ptr;
        }
        return NULL;
    }
};
```

[illegible]

[illegible]

```
class Solution {
public:
    ListNode *detectCycle(ListNode *head) {
        unordered_set<ListNode*> table;
        while(head){
            if (table.find(head) != table.end()) {
                return head;
            }
            table.insert(head);
            head = head->next;
        }
        return NULL;

    }
};
```

21. [illegible]

++

```

    ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
        if (list1 == nullptr) {
            return list2;
        } else if (list2 == nullptr) {
            return list1;
        } else if (list1->val < list2->val) {
            list1->next = mergeTwoLists(list1->next, list2);
            return list1;
        } else {
            list2->next = mergeTwoLists(list1, list2->next);
            return list2;
        }
    }
```

24. [illegible]

[illegible]

1. [illegible]
2. [illegible]
3. [illegible]

++

```
class Solution {
public:
    ListNode* swapPairs(ListNode* head) {
        if (head == nullptr || head->next == nullptr) {
            return head;
        }
        // [illegible]
        ListNode* newHead = head->next;
        // [illegible] newhead [illegible]
        head->next = swapPairs(newHead->next);
        // [illegible]
        newHead->next = head;
        // [illegible]
        return newHead;
    }
};
```

25. K [illegible]

[illegible] *K* [illegible] head[illegible]

[illegible] *K* [illegible]

```
++
class Solution {
public:
    ListNode* reverseKGroup(ListNode* head, int k) {
        ListNode *p = head;
        // [illegible]*K*[illegible] head[illegible]
        // [illegible] K [illegible]
        for(int i = 0; i < k; i++) {
            if(!p) return head;
            p = p->next;
        }

        // *K*[illegible]
        ListNode *q = head;
        ListNode *pre = nullptr;
        while(q != p) {
            ListNode *tmp = q->next;
            q->next = pre;
            pre = q;
            q = tmp;
        }

        // [illegible]
        head->next = reverseKGroup(p, k);
        return pre;
    }
};
```

2. [illegible]

[illegible]

```
[illegible]l1 = [2,4,3], l2 = [5,6,4]
[illegible][7,0,8]
[illegible]342 + 465 = 807.
```

[illegible]

```
++
class Solution {
public:
    ListNode* addTwoNumbers(ListNode* l1, ListNode* l2) {
        ListNode *head = nullptr, *tail = nullptr;
        int carry = 0;
        while (l1 || l2) { //□□□□
            // □□□□□□□□□□□□□□□□□□□□□□□□0
            int n1 = l1 ? l1->val: 0;
            int n2 = l2 ? l2->val: 0;
            // □□□□□□□□□□□□□□
            int sum = n1 + n2 + carry;
            // □□head□□□□□□□□□□□
            // □10□□□□□□□□10□□□□□□□□□□□□□□
            if (!head) {
                head = tail = new ListNode(sum % 10);
            } else {
                tail->next = new ListNode(sum % 10);
                tail = tail->next;
            }
            // □□□□□□
            carry = sum / 10;
            // □□□□□□□□□□□□□□
            if (l1) {
                l1 = l1->next;
            }
            if (l2) {
                l2 = l2->next;
            }
        }
        // □□□□□□□□□□□□□□□□
        if (carry > 0) {
            tail->next = new ListNode(carry);
        }
        return head;
    }
};
```

19. ☒☒☒☒☒☒☒☒ N ☒☒☒

☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ dummy head

```
++
class Solution {
public:
    ListNode* removeNthFromEnd(ListNode* head, int n) {
        if (head == nullptr || head->next == nullptr) {
            return nullptr;
        }
        ListNode* dummy = new ListNode(0, head);
        ListNode* p1 = head;
        ListNode* p2 = dummy;

        for (int i = 0; i < n; i++) {
            p1 = p1->next;
        }

        while(p1){
            p1 = p1->next;
            p2 = p2->next;
        }

        p2->next = p2->next->next;

        ListNode* ans = dummy->next;
        delete dummy;

        return ans;

    }
};
```

146. LRU [illegible]

[illegible]

138. [illegible]

[illegible]

[illegible] hash [illegible] value[illegible]

[illegible]

[illegible] DNA [illegible]

```
++
class Solution {
public:
  unordered_map<Node*,Node*>hmap;
    Node* copyRandomList(Node* head) {
        Node *p=head;
        // [illegible] DNA [illegible]
        // [illegible] value[illegible]
        while(p){
      hmap.insert({p,new Node(p->val)});
```

```
            p=p->next;
    }
        // [illegible]
    p=head;
    while(p){
            // [illegible] next [illegible]
      hmap[p]->next=hmap[p->next];
            // [illegible] random [illegible]
      hmap[p]->random=hmap[p->random];
            p=p->next;
    }
    return hmap[head];
    }
};
```

147. [illegible]

```
++
class Solution {
public:
    ListNode* insertionSortList(ListNode* head) {
        // [illegible]
        if (head == nullptr) {
            return head;
        }

        // [illegible]
        ListNode* dummyHead = new ListNode(0);
        dummyHead->next = head; // [illegible]

        ListNode* lastSorted = head; // [illegible]
        ListNode* curr = head->next; // [illegible]

        // [illegible]
        while (curr != nullptr) {
            // [illegible] lastSorted [illegible]
            if (lastSorted->val <= curr->val) {
                lastSorted = lastSorted->next;
            } else {
                // [illegible]
                ListNode* prev = dummyHead;
                while (prev->next->val <= curr->val) {
                    prev = prev->next;
                }

                // [illegible]
                lastSorted->next = curr->next;
                curr->next = prev->next;
                prev->next = curr;
            }

            // [illegible]
            curr = lastSorted->next;
        }

        // [illegible]
        return dummyHead->next;
    }
};
```

PCIe [illegible]

148. [illegible]

[illegible]head = [4,2,1,3]
[illegible][1,2,3,4]

23. [illegible] K [illegible]

[illegible]

C [illegible] **ctype.h** [illegible]

[illegible]

| int isalpha(int c) | [illegible] | | ——————————————————————— | ————————————— | | int isdigit(int c) | [illegible] | | int islower(int c) | [illegible] | | int tolower(int c) | [illegible] | | int toupper(int c) | [illegible] |

125. [illegible]

```
创创: s = "A man, a plan, a canal: Panama"
创创创true
创创创"amanaplanacanalpanama" 创创创创创
```

[illegible]

[illegible]

```
++
class Solution {
public:
    bool isPalindrome(string s) {
        string temp;
        for (auto c:s) {
            if (isdigit(c)) temp += c;
            if (islower(c))  temp += c;
            if (isupper(c))  temp += tolower(c);
        }
        std::cout<<temp<<std::endl;

        int left = 0;
        int right = temp.length() - 1;
        while (left < right) {
            if (temp[left] == temp[right]) {
                left++;
                right--;
            } else {
                return false;
            }
        }
        return true;

    }
};
```

392. [illegible]

[illegible] `"ace"` [illegible] `"abcde"` [illegible] `"aec"` [illegible]

☒☒☒☒☒☒☒☒☒ i ☒ j☒☒☒☒☒ s ☒ t ☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒☒ i ☒ j ☒☒☒☒☒☒☒ s ☒☒☒☒☒☒☒☒☒☒☒☒ j ☒☒☒i ☒☒☒☒☒☒ t ☒☒☒☒☒ ☒☒☒ s☒

```
++
class Solution {
public:
    bool isSubsequence(string s, string t) {
        int l_s = s.length();
        int l_t = t.length();
        int i = 0, j = 0;

        while(i < l_s && j < l_t) {
            if (s[i] == t[j]) {
                i++;
            }
            j++;
        }
        return i == l_s;

    }
};
```

[illegible]

[toc]

1. [illegible]

```
[illegible]nums = [2,7,11,15], target = 9
[illegible][0,1]
[illegible] nums[0] + nums[1] == 9 [illegible] [0, 1] [illegible]
```

[illegible]

++
```
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {
        unordered_map<int ,int> table;
        for (int i = 0; i < nums.size(); i++) {
            auto it = table.find(target - nums[i]);
            if (it != table.end()) {
                return {it->second, i};
            }

            table[nums[i]] = i;

        }

        return {};

    }
};
```

49. [illegible]

```
[illegible]: strs = ["eat", "tea", "tan", "ate", "nat", "bat"]
[illegible]: [["bat"],["nat","tan"],["ate","eat","tea"]]
```

++
```
class Solution {
public:
    vector<vector<string>> groupAnagrams(vector<string>& strs) {

        // [illegible]key [illegible]value [illegible]
        unordered_map<string, vector<string>> map;

        // [illegible]
        for (auto str:strs){
            string key = str;
            sort(key.begin(), key.end());
            map[key].push_back(str);
        }

        // □□□□□□□□□□□□□□□□ ans □
        vector<vector<string>> ans;
        for (auto it = map.begin(); it != map.end(); it++) {
            ans.push_back(it->second);
        }
        return ans;

    }
};
```

128. □□□□□□

□□□nums = [100,4,200,1,3,2]
□□□4
□□□□□□□□□□□□ [1, 2, 3, 4]□□□□□□ 4□

```
++
class Solution {
public:
    int longestConsecutive(vector<int>& nums) {
        // □□□□□□□, □□□□□□□□
        unordered_set<int> num_set;
        for (const int& num : nums) {
            num_set.insert(num);
        }

        int longestStreak = 0;

        for (const int& num : num_set) {
            // □□ num - 1 □□ num_set □□□□ num □□□□□□□□□□□
            if (!num_set.count(num - 1)) {
                int currentNum = num;
                int currentStreak = 1;

                // □ num □□□□□□□□□□□□□□□□□□□□□□□□□□□
                // □□□□□□□□□□□ currentStreak□
                while (num_set.count(currentNum + 1)) {
                    currentNum += 1;
                    currentStreak += 1;
                }

                // □□ max □□□□ longestStreak□
                // □□□□□□□□□□□□□□□
                longestStreak = max(longestStreak, currentStreak);
            }
        }

        return longestStreak;
    }
};
```

[illegible]

[toc]

283. [illegible]

[illegible] nums[illegible] 0 [illegible]

```
[illegible]: nums = [0,1,0,3,12]
[illegible]: [1,3,12,0,0]
```

[illegible] 0[illegible] 0[illegible] 0[illegible]

```
++
class Solution {
public:
    void moveZeroes(vector<int>& nums) {
        int n = nums.size(), left = 0, right = 0;
        while (right < n) {
            if (nums[right]) {
                swap(nums[left], nums[right]);
                left++;
            }
            right++;
        }
    }
};
```

11. [illegible]

```
++
class Solution {
public:
    // [illegible]
    int maxArea(vector<int>& height) {
        int left = 0, right = height.size() - 1;  // [illegible]
        int maxArea = 0;  // [illegible] 0

        // [illegible]
        while (left < right) {
            // [illegible]
            int currentArea = (right - left) * min(height[left], height[right]);
            // [illegible]
            maxArea = max(maxArea, currentArea);

            // [illegible]
            if (height[left] < height[right]) {
                left++;
            } else {
                right--;
            }
        }

        return maxArea;  // 创创创创创创
    }
};
```

15. □□□□□

[illegible]

[toc]

20. [illegible]

[illegible] '(' [illegible] ')' [illegible] '{' [illegible] '}' [illegible] '[' [illegible] ']' [illegible] s [illegible]

[illegible] ([{]}) [illegible]

[illegible]

++
```
class Solution {
public:
    bool isValid(string s) {
        int n = s.size();
        if (n % 2 == 1) {  // [illegible] false
            return false;
        }

        // [illegible]
        unordered_map<char, char> pairs = {
            {')', '('},
            {']', '['},
            {'}', '{'}
        };

        stack<char> stk;  // [illegible]

        // [illegible]
        for (char ch: s) {
            if (pairs.count(ch)) {  // [illegible]
                // [illegible] false
                if (stk.empty() || stk.top() != pairs[ch]) {
                    return false;
                }
                stk.pop();  // [illegible]
            } else {
                stk.push(ch);  // [illegible]
            }
        }

        return stk.empty();  // [illegible]
    }
};
```

155. [illegible]

[illegible] push [illegible]pop [illegible]top [illegible]

[illegible]

[illegible]

```
++
class MinStack {
public:
    stack<int> x_stack;
    stack<int> min_stack;

    MinStack() {
        min_stack.push(INT_MAX);
    }

    void push(int val) {
        x_stack.push(val);
        min_stack.push(min(min_stack.top(), val));
    }

    void pop() {
        x_stack.pop();
        min_stack.pop();
    }

    int top() {
        return x_stack.top();
    }

    int getMin() {
        return min_stack.top();
    }
};

/**
 * Your MinStack object will be instantiated and called as such:
 * MinStack* obj = new MinStack();
 * obj->push(val);
 * obj->pop();
 * int param_3 = obj->top();
 * int param_4 = obj->getMin();
 */
```

394. [illegible]

```
[illegible]s = "3[a]2[bc]"
[illegible]"aaabcbc"
```

++
```
string decodeString(string s) {
    string ans = "";  // [illegible]

    stack<int> num_stk;     // [illegible]
    stack<string> str_stk;  // [illegible]

    int temp_num = 0;       // [illegible]
    int len_s = s.size();   // [illegible]

    for (int i = 0; i < len_s; i++) {
        if (isdigit(s[i])) {
            temp_num = temp_num * 10 + s[i] - '0';  // [illegible]
        } else if (isalpha(s[i])) {
            ans = ans + s[i];  // [illegible]
        } else if (s[i] == '[') {
            num_stk.push(temp_num);     // [illegible]
            temp_num = 0;               // [illegible] temp_num
            str_stk.push(ans);          // [illegible] ans [illegible]
            ans = "";                   // [illegible] ans[illegible]
        } else if (s[i] == ']') {
            int times = num_stk.top();      // [illegible]
            num_stk.pop();                  // [illegible]
            string temp = str_stk.top();    // [illegible]
            str_stk.pop();                  // [illegible]

            // [illegible] ans [illegible] times [illegible] temp [illegible]
            for (int j = 0; j < times; ++j) {
                temp += ans;
            }

            ans = temp;  // [illegible] ans [illegible]
        }
    }

    return ans;  // [illegible]
}
```

739. [illegible]

```
[illegible]: temperatures = [73,74,75,71,69,72,76,73]
[illegible]: [1,1,4,2,1,1,0,0]
```

[illegible]

++

```
vector<int> dailyTemperatures(vector<int>& temperatures) {
    int n = temperatures.size();  // [illegible]

    vector<int> ans(n, 0);  // [illegible]0
    stack<int> st;  // [illegible]

    for (int i = 0; i < temperatures.size(); i++) {
        // [illegible]
        while (!st.empty() && temperatures[i] > temperatures[st.top()]) {
            auto t = st.top();  // [illegible]
            st.pop();  // [illegible]
            ans[t] = i - t;  // [illegible]
        }
        st.push(i);  // [illegible]
    }

    return ans;  // [illegible]
}
```

System Design

☒☒☒☒

☒☒☒☒☒☒☒

system-design-primer

System Design Overview

Tool

Tool

Git

[illegible] git bash [illegible] git status [illegible]

[illegible] git [illegible] `core.quotepath` [illegible] false[illegible] quotepath [illegible] [illegible]`--global` [illegible]

git bash [illegible]

```
git config --global core.quotepath false
```

☒☒

[illegible]

[toc]

[illegible]

[illegible]

1. [illegible]
2. [illegible]
3. [illegible]
4. [illegible]
5. [illegible]
6. [illegible]
7. [illegible]

[illegible] Meeting Minutes

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

1. [illegible]
2. [illegible] Jira [illegible] HSD [illegible] unit test[illegible]
3. [illegible] BIOS [illegible] OS[illegible] BIOS [illegible]

One on One

[illegible] manager[illegible] manager[illegible] second manager[illegible]arch[illegible]Jun [illegible] one one[illegible]

2024 Reading

[toc]

[illegible] 1566

[illegible]——[illegible]

[illegible]

[illegible]

[illegible]——[illegible] GPT-4 [illegible] ChatGPT

ARM [illegible]

[illegible]