Department of Electronic and Telecommunication Engineering

University of Moratuwa, Sri Lanka

EN2030 - Fundamentals of Computer Organization and Design

PROCESSOR DISSECTION REPORT

Group 32

Submitted by

SIRITHUNGA M.R.A.	180609B
SOMARATHNE P.M.P.H.	180616T
THALAGALA B.P.	180631J

Submitted on

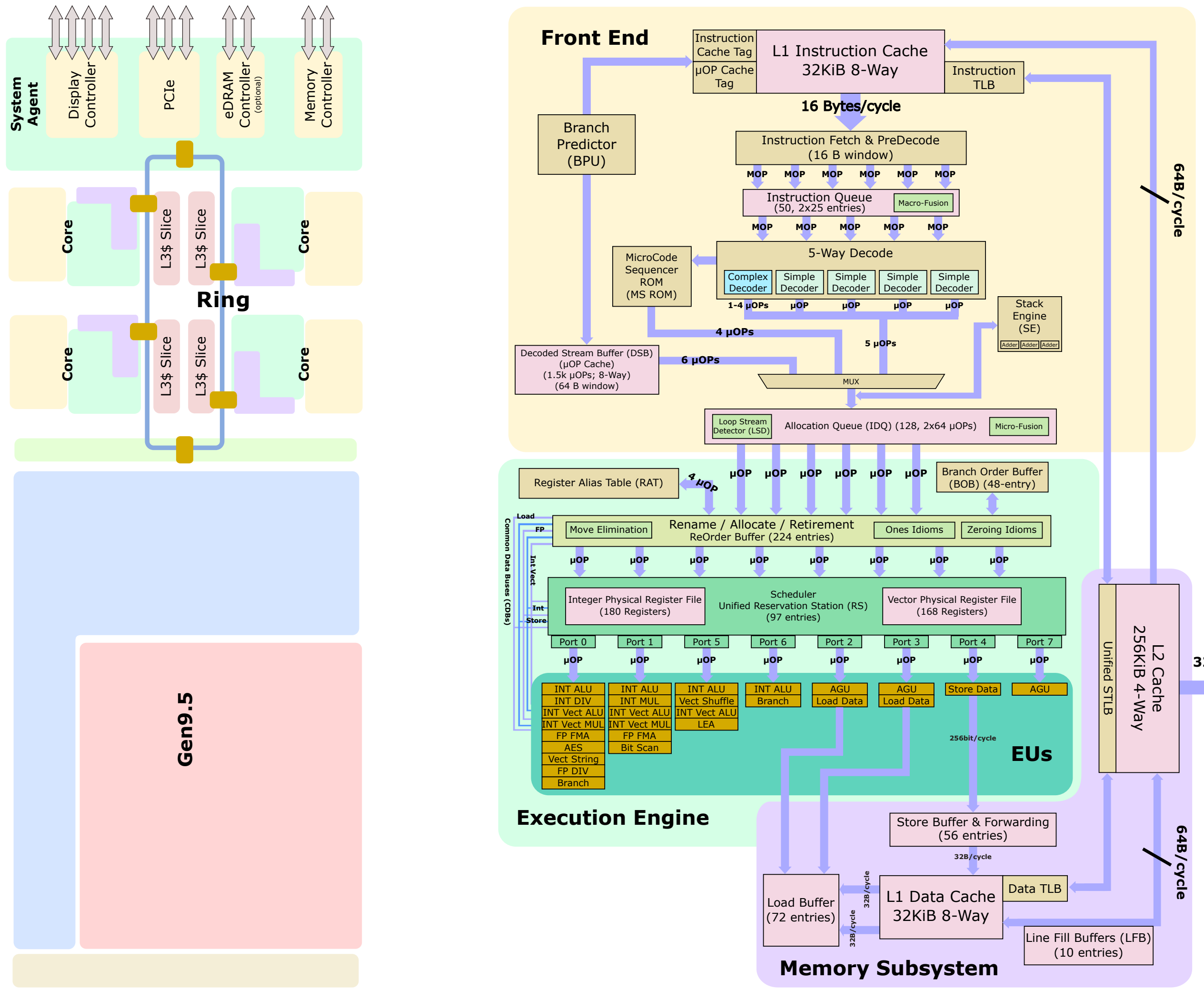
November 12, 2020

Contents

Contribution from each member to the project

Index	**Name**	**Contribution(Sections Covered)**
180609B	SIRITHUNGA M.R.A.	* Micro-Architecture (Data Path and the Controller) * ALU functions
180616T	SOMARATHNE P.M.P.H.	* Instruction Set * Instruction classes and Instruction Format
180631J	THALAGALA B.P.	* Cache Memory and Memory Interfacing * Timing related to Memory

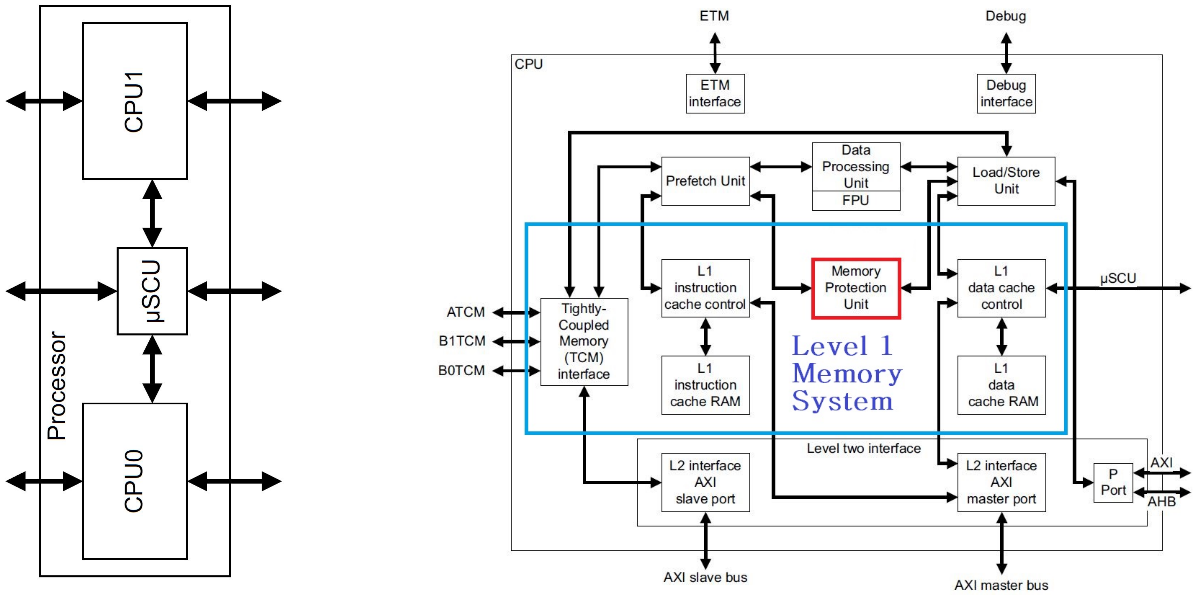
Table 1: Contribution from each member to the project



(a) Quad Core Core-i3 8300 Processor

(b) A Single Core

Figure 1: Intel core-i3 8300 Processor



(a) Dual Core - Cortex -R5 Processor

(b) A Single CPU

Figure 2: ARM Cortex R5 Processor

1 Instruction Set Architecture of the Processor

1.1 Instruction Set

1.2 Instruction classes and Instruction Format

glghuly

2 Micro-Architecture (Data Path and the Controller)

Computer architecture consists of two main branches called instruction set architecture (ISA) and microarchitecture (μarch). A given ISA can be implemented with different microarchitectures. So, microarchitecture depends on the ISA and the technologies used in implementations. The microarchitecture is the digital logic that defines the way of how the instructions should be executed. Here we have focused on the organization of the Data path and the controller of the internal processor design.

2.1 Arm Cortex-R5 Processor

This is a mid-range CPU which is widely used in embedded, real-time systems. To that, ARMv7-R architecture is implemented in cortex-R5.ARM is a RISC architecture. Microcontroller Bus architecture has been embedded with it for better performance. When it comes to the internal design of the processor, the processor may have single or dual CPU configurations. However, the CPU data path is common at all 32bits or 64bits data paths will be used.(Refer figure 2(b))

The PreFetch Unit (PFU) fetches instructions from the memory device, forecasts branches, and forwards instructions to the Data Processing Unit (DPU). Both directions are executed and the Load / Store Unit (LSU) is used for transferring data memory. The L1 interface, which includes L1 instructions and caches and TCM interfaces, is supported with the PFU and LSU interface. The L1 caches are in turn connected to the L2 memory system and the LSU is connected by a more direct peripheral port with the L2 memory system. For cache management needed to be coherent with ACP transactions, the Data Cache interfaces L1 to the μSCU.

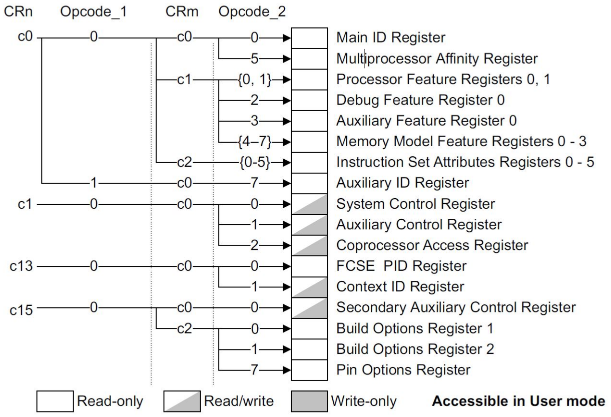


Figure 3: System control and configuration registers

There is a system control coprocessor as the controller in cortex-R5 implementation. All control signals that manipulate the registers, system-level operations, cache maintenance, and such memory system functionalities are produced by this. There are 18 read-only registers and 7 read/write registers among them as shown here.

2.2 Intel Core i3-8300 Processor

The internal design of an intel core i3 8300 is very advanced and complicated. It is CISC based microarchitecture called 'Coffee Lake'. The actual size of a data path is 64 bits. Three levels of caches have been implemented inside it for better performance. Uni-processor configuration is used with four CPU cores and threads in the design. This is pipelined. The focus of the pipelining is to reduce power consumption and enhance overall performance. The coffee lake's pipeline is the same as intel's sky lake pipelining strategies. So, it is embedded in additional parallelism for achieving its performance. The pipeline can be broken down into three zones: the ***front-end***, ***back-end or execution driver***, and the ***memory subsystem***. The front-end aims to feed the back end with an adequate stream of operations that it collects by decoding instructions from memory. The front-end has two primary pathways: the micro-operations(μOPs) cache route and the legacy path. The legacy path is the standard path through which variable-length x86 instructions are extracted from the Level 1 instruction cache, queued, and subsequently decoded into shorter, fixed-length μOPs. The alternative and much more suitable approach is the μOPs cache path through which a cache containing already decoded μOPs receives a hit causing the μOPs to be transferred directly to the decode list.

In the back end, a micro-operation visits the reorder buffer. It is where the allocation, renaming, and deletion of the registry takes place. A variety of other optimizations are also performed at this point. The μOPs are sent from the reorder buffer to the unified scheduler. The scheduler has a variety of escape ports, each of which is attached to a series of separate execution units. Some units can execute simple ALU operations, others can multiply and split, with some units capable of more complex operations, such as separate vector operations. The scheduler is essentially responsible for queuing the μOPs to the appropriate port so that they can be performed by the appropriate device.Any μOPs deal with access to memory, load & store. Those will be sent to the dedicated scheduling ports that can handle the memory operation. Store operations go to the store buffer, which is also capable of forwarding as required. Load operations often come from a load buffer. According to that, the internal design consists of three major sub designs. These are focused on the following improvements,

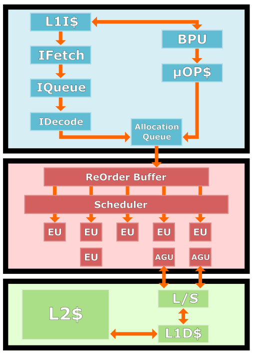


Figure 4: μarch: of Intel Core-i3 8300

1. **Front end**

   I. Increase the legacy pipeline delivery to 5 microoperations.

   II. Increase the IDQ delivery to 6 microoperations.

   III. Support 2.28x larger allocation queue that has 64/thread.

   IV. Improves the performance of the branch prediction unit.

2. **Execution engine**

   I. Increase the re-order buffer to 224 entries.

   II. Increase the scheduler to 97 entries and the integer register file to 180 entries.

   III. Increase the store buffer to 56 entries.

3. **Memory subsystem**

   I. 8-way to 4-way set associative mapping.

The fetching and decoding of the instructions are happening separately. Fetched instructions are stored in a queue (FIFO). After decoding, these are executed in the execution engine. All these three subsystems are controlled by the control signals. Generating control signals are done in a control store.

---

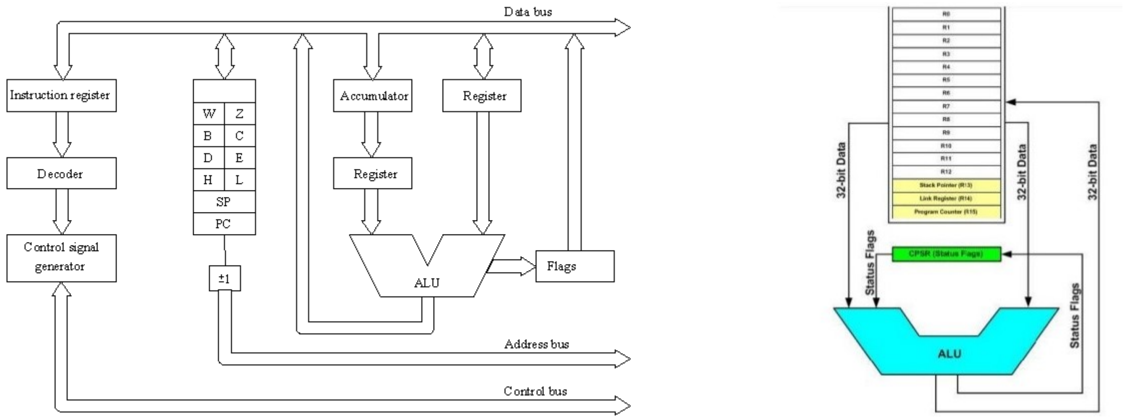
3 ALU functions

The arithmetic and logic unit is abbreviated as ALU in the computer organization. ALU is responsible for all arithmetic and logical operations. It is implemented by using digital logic for the respective set of operations that are required.

3.1 Arm Cortex-R5 Processor

In the eight-stage pipeline, an ALU performs its task. It two data paths that are directly connected with 32 bits registers. Cortex-R5 is well designed for digital signal processing and efficiently performed floating-point number calculations.(figure 5(b)) Four flags stand for,

- N - Negative
- Z - Zero the ALU output subjected to NOR operation
- C - Carry the Count bit
- V - Overflow V bit output



(a) Core-i3 8300 Processor's ALU

(b) Cortex-R5 processors's ALU

Figure 5: ALUs of the two processors

3.2 Intel Core i3-8300 Processor

Intel's Coffee Lake architecture is used 32bits registers. So, for that ALU should be capable of manipulating these numbers in it. This is implemented by using digital logic.(figure 5(a)) Two of them are embedded in a processor for,

- Physical address calculations
- Mathematical operations

Core i3 processors are widely used in desktop and laptop general-purpose computers. The clock frequency is around 3.7GHz so, Its ALU is running at a higher speed than its clock rate.

4 Cache Memory and Memory Interfacing

4.1 Intel Core i3-8300 Processor

4.1.1 Memory Hierarchy

Cache Memory Intel core i3-8300 processor's cache memory is organized using ***Intel's Smart Cache*** technology. That is the Last Level Cache(LLC) is shared across all cores while the lower level caches are separately allocated for each core such that those are used by the respective cores privately. The shared cache allows any of the four cores to access the entire storage area of the shared cache(*in this case 8MiB LLC*) and therefore not limited to a dedicated portion of it. This leads to a number of benefits such as, increased resource utilization through providing all of the shared cache to the active cores if the other cores are in the idle mood, reduced front-side bus traffic since shared data can be fetched directly from the LLC(*if available*) into the cores rather than going all the way to the primary memory[6].

Following table summarizes the properties of each cache. Refer the *memory subsystem* of the figure 1(b) to identify the organization of core-private caches while shared cache(LLC/L3$) is depicted inside the *Ring* of the figure 1(a).

Consider these abbreviations to refer tables, D-*Data*, I-*Instruction*, WB- *Write-back*, WT- *Write-through* U-*Unified*, S-*Shared*, SA-*Set Associative*

Cache	Mapping Technology	Cache Size	No: of Sets	Cache line size	Writing Policy
L0 µOP Cache	8-way SA	1,536 µOPs	32	6-µOP	N/A
L1 I Cache	8-way SA	32 KiB	64	64 B	N/A
L1 D Cache	8-way SA	32 KiB	64	64 B	WB
L2 U cache	4-way SA	256 KiB	1024	64 B	WB
L3 U, S Cache/LLC	Up to 16-way SA	8 MiB	8192	64 B	WB

Table 2: Cache Memory Organization in Intel core i3-8300 Processors[2]

Primary/Physical/Main Memory The next lower level memory after the L3 cache(LLC) is the System DRAM(Dynamic Random Access Memory) which is also known as the Primary/Main/Physical memory. Intel processors come in 4 different memory channel configurations as, *Single channel, Dual channels, Triple channels and Flex mode.* Intel core i3-8300 Processor is Dual channel and has the capability of reading from or writing to the primary memory in a maximum rate of 37.5GB/s. Moreover the processor supports upto 64GB of DDR4-2400 RAMs(4th generation of Double Data Rate(DDR) RAMs with 2400 Mbps data transfer rate) which is also an ECC(Error-Correcting Code) memory with the ability of detecting and correcting of common types of internal data corruptions.

4.1.2 Translation-Lookaside Buffers(TLBs)

In a ***Virtual Memory System Architecture***(VMSA)(technique of using primary memory as a cache for the secondary memory) the processor generates *virtual addresses* while the memory is accessed using the *physical addresses.* The mechanism of translating a virtual address to a physical address is called ***Address Translation/ Mapping*** and it consumes time. Because a single memory access in such a virtual system is actually a two physical memory accesses: first access to obtain the physical address corresponding to the virtual address from the *page table*(part of the memory where physical addresses corresponding to virtual addresses are stored) and the second access to obtain the required data stored in that physical address.

To reduce this latency, an address-translation cache which is known as ***Translation-Lookaside Buffer(TLB)*** where recent address translations are stored is used. In a single core of the Intel core i3-8300 Processor there are three TLBs and their properties are given in the following table while the physical placement is shown in the *Memory Subsystem* and the upper part of the *Front End* of the figure 1(b).

TLB	Mapping Technology	Page size	No: of Entries	Partitioning Method
I-TLB	8-way SA	4 KiB	128	Dynamic
D-TLB	4-way SA	4 KiB	64	Fixed
STLB U	12-way SA	4 KiB + 2 MiB	1536	Fixed

Table 3: TLBs Organization in Intel core i3-8300 Processors

4.1.3 Store Buffers

Each core of an Intel Core-i3 processor consists of a Store Buffer which is located between the *Port 4*(dedicated port for storing data) of the scheduler and the *L1-Data cache* as shown in the figure 1(b). Every ***memory write*** operation carried out by the processor is temporarily stored in this buffer before they are executed. So the processor does not have to wait until the operation is finished and it can carry out the rest of the instructions freely. This mechanism increases the processor's overall performance through ***eliminating the unwanted idling***.

4.2 Arm Cortex-R5 Processor

4.2.1 Memory Hierarchy

Cache Memory Arm Cortex-R5 Processor's CPUs only have Level 1(L1) integrated cache controllers while ARM-L2 cache controllers can be connected outside of the processor instance according to the requirement, by the system designer. L1 caches are split as *L1 Instruction cache* and *L1 Data cach*e in order to increase the performance through increasing the bandwidth. Moreover their sizes can be independently configured to be between 4KB and 64KB and each cache can be disabled independently. Data and Instructions are fetched in to the L1 caches via the *AXI master port* at Level 2 Interface(shown in figure 2(b)) from the external memory.

When considering the cache organization of L1 caches, they are always implemented using the ***Set Associative Mapping Technology*** in order to reduce the cache thrashing(loosing of data in a cache line at a given index, when replacing it with a new cache line with the same index) come across in the Direct Mapping Technology. One cache line consists of ***8-words*** and ***Pseudo-random cache replacement policy*** is used to *randomly select* a cache line to be replaced in a given *set*(a group of cache lines with the same index) for an incoming new cache line at the occurrence of a *cache miss*. Moreover the ***critical word is first filled***(the word requested by the processor) to the cache line at such a cache miss, rather than waiting for the whole memory block in order to increase performance. Additionally the writing policy can be configured using the Memory Protection Unit(MPU) to be ***either write-back or write-through***.

Tightly-Coupled Memories (TCMs) *Unpredictable access time* at a processor request is a common issue related to the cache memories because, access times at a *cache hit* and a *cache miss* are different in nature. Tightly-Coupled Memories (TCMs) address this issue and provide ***low-latency*** memory access and ***consistent access time*** which is ideal for storing time-critical routines[1]. Cortex-R5 processor can be configured to have one or two TCMs(ATCM & BTCM) which are located separately from the processor. They may contain any mix of Data and Instructions and can have capacities upto 8MB. These TCMs can be implemented as SRAMs or ROMs and typically fetch data in a single cycle. ATCM has a single port while BTCM has two ports which can be accessed simultaneously as it is implemented as two banks of RAMs. TCMs can be loaded via the *AXI slave port* at Level 2 Interface(shown in figure 2(b)) and processor can fetch instructions from the TCMs directly without going all the way to an external memory. Although the TCM is implemented as a separate RAM it does not have a dedicated *address space*(set of addresses) and it simply ***uses a portion of the 32-bit address space*** which is used by the processor at normal operation. Therefore when these TCMs are enabled anything at the same address space(as that is allocated for the TCMs) in the external memory is not accessible to the processor[3].

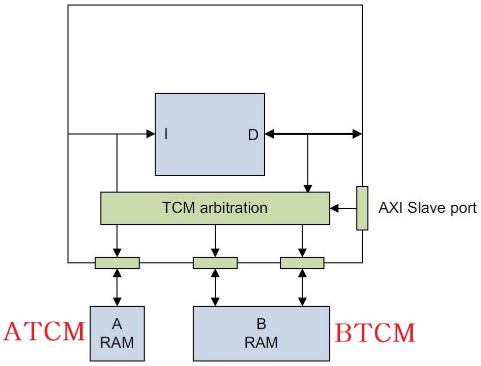


Figure 6: TCMs in Cortex-R5

4.2.2 Memory Protection Unit(MPU)

In a multitasking system, the task currently being executed must not affect the system resources(code, data) of the other tasks. This protection mechanism is controlled by the Operating System(OS), typically with the help of both hardware and software. In ARM Cortex-R series processors which implement the ARM ***Protected Memory System Architecture***(PMSA) this feature is provided by a dedicated hardware called *Memory Protection Unit*(MPU)[1] located inside the *Level 1 Memory System*(shown in figure 2(b)) of the cores. Using the MPU, memory can be partitioned into ***zero, 12 or 16 regions*** and protection attributes can be set for each region independently. The size of such a region is specified by a 5-bit value which encodes a range of values from 32-Bytes(cache-line length) to 4GB.[3]

5 Timing related to Memory

5.1 Intel Core i3-8300 Processor

In coffee lake μarchitecture the processor is divided into several ***Clock Domains*** where each part maintains a different clock frequency which is applicable only to that part. All of these frequencies are some multiple of what is known as *Base Clock* which acts only as a reference for other clock domains[5] . Cache memories, their related clock domains, latency and bandwidths are as follows. In addition to that system's DRAMs are operated under the *Memory clock* domain and are capable of transferring data at the rate of 8 bytes per cycle per channel[2](Core i3-8300 Processor is dual channel as mentioned previously).

Cache	Clock Domain	Fastest Latency (cycles)	Peak Bandwidth (bytes/cyc)	Sustained Bandwidth (bytes/cyc)
L1 I Cache	Core Clock	N/A	N/A	N/A
L1 D Cache	Core Clock	4	96	81
L2 U cache	Core Clock	12	64	29
L3 U, S Cache/LLC	Ring Clock	44	32	18

Table 4: Timing Related to Cache Memory in Intel core i3-8300 Processors[4]

5.2 Arm Cortex-R5 Processor

One Thumb instruction can be fetched on each cycle, whereas each 32-bit ARM instruction requires two clock cycles per fetch.
In many ARM processor-based systems, access to external memory will take tens or even hundreds of core cycles
CLKIN (Input) Master processor clock.
2.3.3 Clocking

6 Comaprison

Feature	ARM cortex-R5	Intel Core-i3-8300
ISA	RISC	CISC
Memory Sys: Archi:	Protected Mem: Sys: Archi:(PMSA)	Virtual Mem: Sys: Archi:(VMSA)

Table 5:

6.1 Instruction Set Architecture of the Processor

6.2 Micro-Architecture (Data Path and the Controller)

6.3 ALU functions

6.4 Cache Memory and Memory Interfacing

6.5 Timing related to Memory

Bibliography

[1] Arm cortex-r series programmer's guide. https://developer.arm.com/documentation/den0042/a/.

[2] Coffee Lake - Microarchitectures - Intel - WikiChip. https://en.wikichip.org/wiki/intel/microarchitectures/coffee_lake.

[3] Cortex-r5 technical reference manual. https://developer.arm.com/documentation/ddi0460/d.

[4] Intel® 64 and IA-32 Architectures Optimization Reference Manual. https://www.intel.com/content/www/us/en/develop/download/intel-64-and-ia-32-architectures-optimization-reference-manual.html.

[5] Skylake (client) - Microarchitectures - Intel - WikiChip. https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(client).

[6] Software Techniques for Shared-Cache Multi-Core Systems. https://www.intel.com/content/www/us/en/develop/articles/software-techniques-for-shared-cache-multi-core-systems.html.