

# DreamCore

李志锐

# 报告内容

- 简介
- 架构篇
- 工程Flow与工具篇

# 简介

# 简介

- DreamCore现拥有两个版本：
- V1（2022.01~2022.09），V2（2022.10~Now）
- 跑分：

	V1	V2（乱序LSU实现前）
Coremark GCC7.2	4.60/MHz IPC: 1.23 BHR: 86.48%	4.99/MHz IPC: 1.34 BHR: 90.78%
Coremark GCC8.3	4.47/MHz IPC: 1.26 BHR: 87.70%	12.2: 4.83/MHz IPC: 1.30 BHR: 90.75%
Dhrystone GCC7.2	2.25DMIPS/MHz IPC: 1.61 BHR: 96.30%	2.56DMIPS/MHz IPC: 1.54 BHR: 99.05%
Dhrystone GCC8.3	3.06DMIPS/MHz IPC: 1.84 BHR: 99.99%↓	12.2: IPC: 1.48 BHR: 99.34%

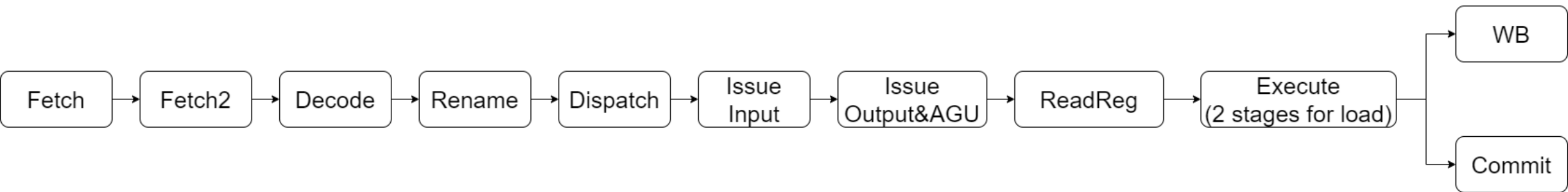
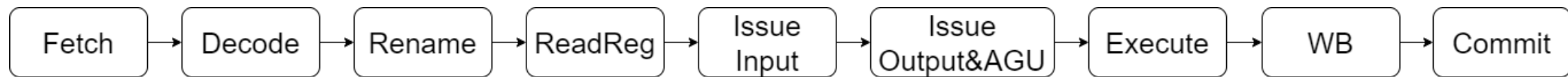
- V1 FPGA综合（VCU1525（xcvu9p-fsgd2104-2L-e）：10MHz收敛）
- LUT: 559892 DFF: 478328 BRAM: 512

## V1与V2特点

- RV32IM指令集
- 9级流水乱序超标量（CSR与LSU指令不支持乱序）
- SystemVerilog全参数化设计
- 包含与RTL每周期行为完全一致的Cycle Accurate Model及GUI Debugger可以观测和控制微架构行为
- 前端（取指、译码、重命名、寄存器读取）宽度为4发射宽度为2
- 写回宽度为8
- 提交宽度为4
- 退休宽度为4
- 16项压缩型发射队列
- 64项ROB
- 256项Checkpoint
- GShare+Local混合预测器（4K项PHT, 12bit历史记录），另有两个4K的BHT负责call型及普通（如跳转表）间接跳转指令的地址预测支持（与方向预测器共用GHR）
- 16项Store Buffer
- 1MB 2R1W TCM
- Clint及UART Controller（256 TX/RX FIFO）

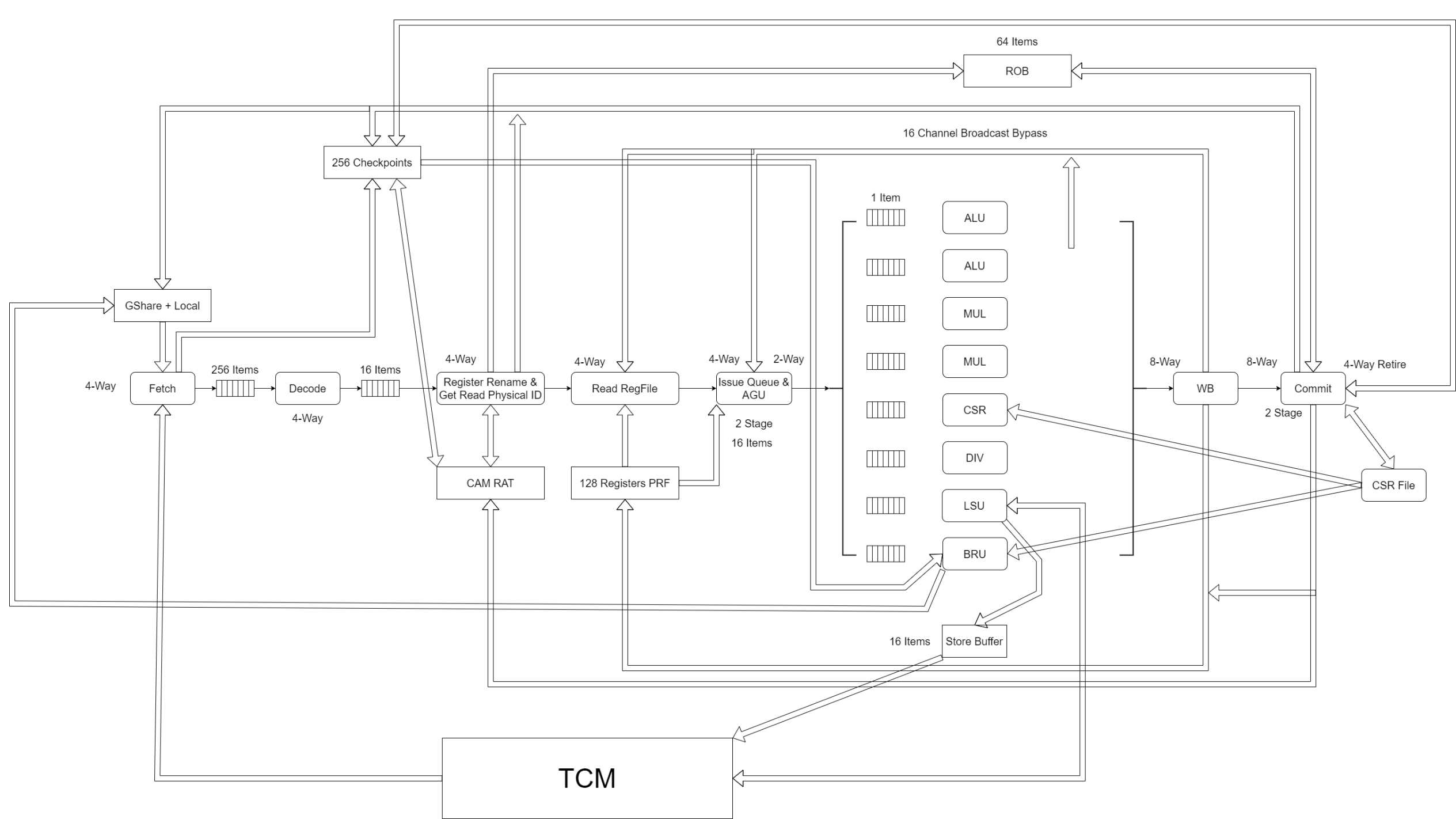
- RV32IM指令集
- 12级流水乱序超标量（CSR指令不支持乱序，LSU指令全乱序，后续可能支持多路LSU）
- SystemVerilog全参数化设计
- 包含与RTL每周期行为完全一致的Cycle Accurate Model及GUI Debugger可以观测和控制微架构行为
- 前端（取指、译码、重命名、分发）宽度为4
- 整数发射队列发射宽度为2，LSU发射队列发射宽度为3（仅一条load、一条store addr、一条store data，发射时store指令自动拆解为store addr与store data两条uop）
- 整数与LSU寄存器读取宽度为3
- 写回宽度为9（最多只有4个有效项，3个整数，1个LSU）
- 提交宽度为11（最多只有6个有效项，3个整数，3个LSU）
- 退休宽度为4
- 16项非压缩型Oldest整数和LSU发射队列（支持提前唤醒、推测唤醒和Replay，LPV相关分析方法）
- 64项ROB与Checkpoint，且uop共享同一个ROB项
- Bi-Modal（PHT 16项）与Bi-Mode（PHT 64K项，16bit历史记录与16bit分支地址）两级方向预测器（后续考虑换TAGE），L0\_BTB（16项）与L1\_BTB（暂未实现）两级间接跳转预测器
- 16项Store Buffer及32项Load Queue
- 1MB 2R1W TCM
- Clint及UART Controller（256 TX/RX FIFO）

# V1与V2的流水线结构对比



# 架构篇

V1 架构



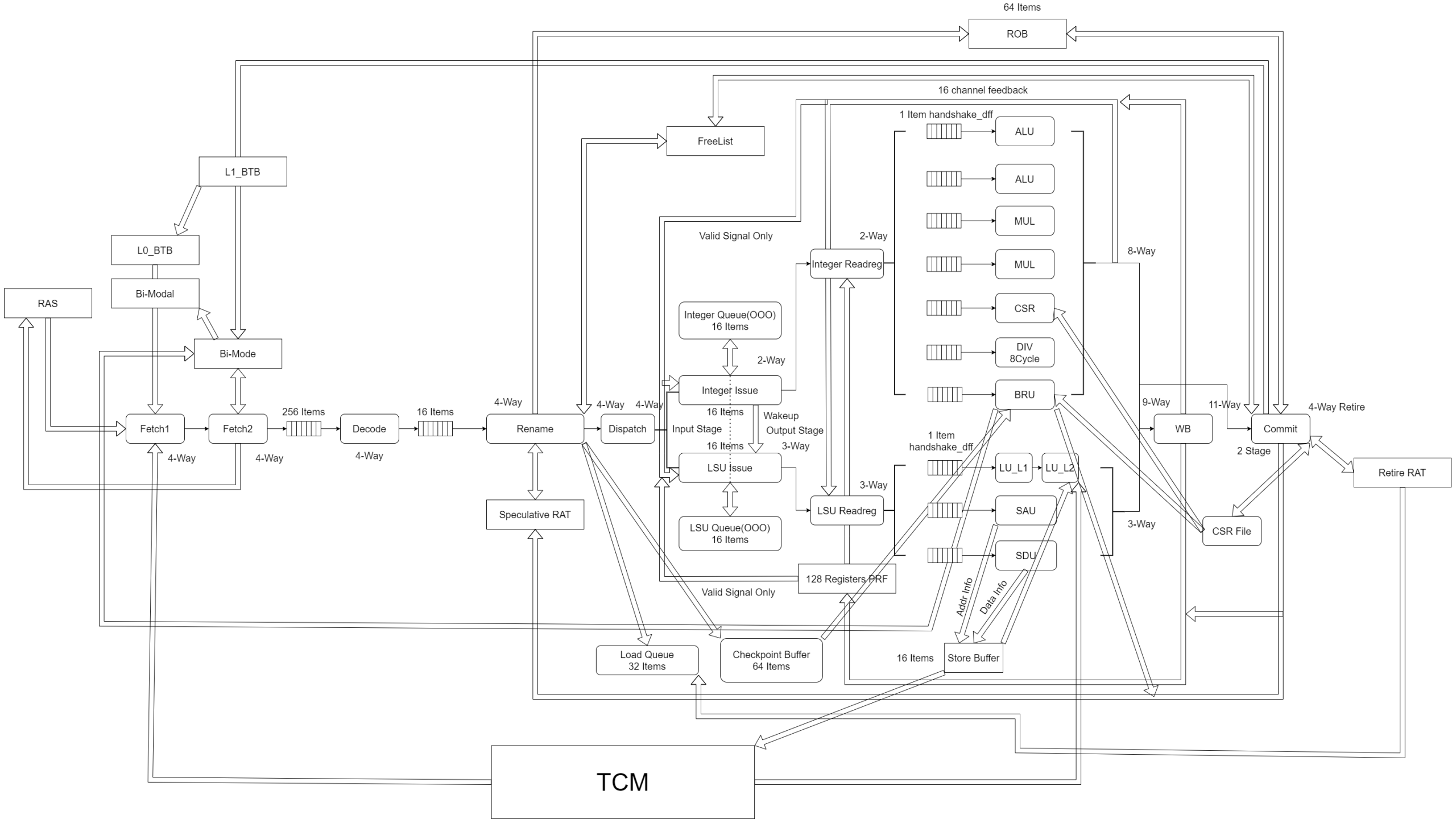


# 架构存在的问题及解决方案（V2引入）

- Rename寄存器选择算法逻辑过于复杂（60~70ns） - 加入FreeList
- 除法器单周期逻辑延迟过大（50ns） - 变为多周期或多级流水
- 将Checkpoint用于分支指令退休时的恢复是不合理的，且无法令中断异常进行快速恢复 - 使用Speculative RAT + Commit RAT结构
- Regfile端口数量过大可能会带来一定的频率影响 - RF<->Issue流水级对调
- 数据捕捉压缩型发射队列对频率影响较大 - 改为非数据捕捉非压缩型发射队列
- Issue流水级的反馈检测逻辑和发射仲裁逻辑加起来路径过长（19ns，大概是7+12ns，其中7ns是rob->commit->execute\_mul->issue反馈路径的长度） - 拆分为wakeup与issue output两个流水级
- 执行单元到issue output级的繁忙信号组合逻辑过长 - 在issue output级使用倒计数器计算执行单元空闲情况，去掉反馈回路
- AGU嵌入issue output级带来几个ns的延迟影响 - 将地址计算延迟到LSU中做，并将LSU拆分为两个流水级，分别负责地址计算和数据获取

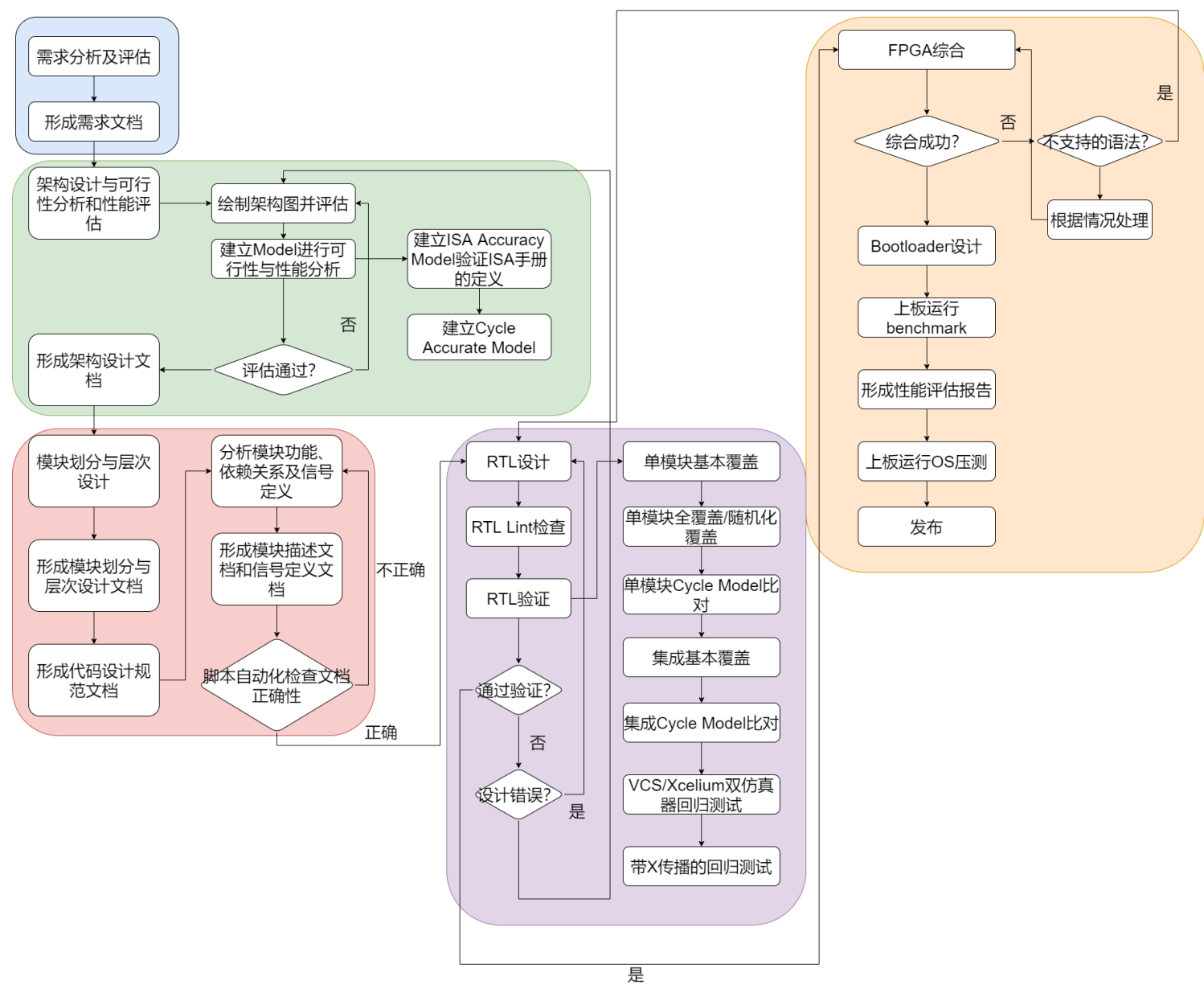
# 架构篇

V2 架构



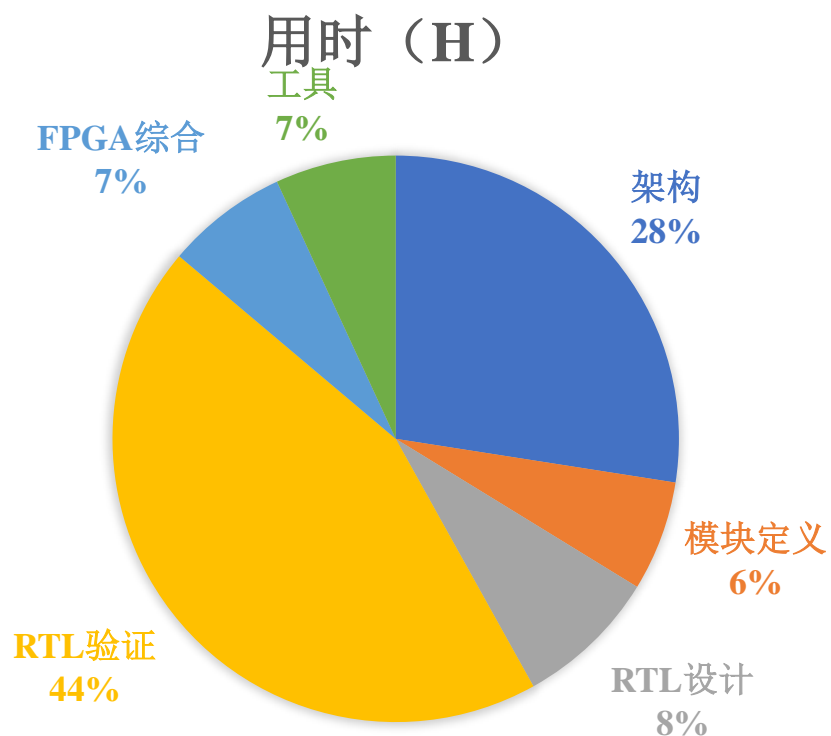
# 工程Flow与工具篇

# 该处理器设计所采用的工程Flow



# DreamCore V1设计时间

- 总时间：668h（按一天8h计，共83.5天）
- 架构：183.5h 模块定义：42h RTL设计：54.5h RTL验证：295.5h
- FPGA综合：46.5h 工具：46h 软件移植：忽略不计



# 工具

- C++17编写Model, C# .net 6.0 WPF编写Model Debugger
- Python的RTL生成与多核仿真脚本
- VCS/Verdi 2018.09与Xcelium/Indago 20.09双仿
- 一套Trace数据库生成与读取库, 生成库面向C++, 读取库面向SystemVerilog, 可以将每个周期的Model信号全部保存到数据库中, 并在SV端比对时读取
- Vivado 2021.2

Address			Data	Instruction	Register	Value
80000040	93 07 00 00		mv a5, zero		x0/zero	0x00000000(0)
80000044	13 08 00 00		mv a6, zero		x1/ra	0x00000000(0)
80000048	93 08 00 00		mv a7, zero		x2/sp	0x80017954(2147580244)
8000004C	13 09 00 00		mv s2, zero		x3/gp	0x8000F928(2147547432)
80000050	93 09 00 00		mv s3, zero		x4/tp	0x00000000(0)
80000054	13 0A 00 00		mv s4, zero		x5/t0	0x800103E0(2147550176)
80000058	93 0A 00 00		mv s5, zero		x6/t1	0x80017954(2147580244)
8000005C	13 0B 00 00		mv s6, zero		x7/t2	0x00000000(0)
80000060	93 0B 00 00		mv s7, zero		x8/s0/fp	0x00000000(0)
80000064	13 0C 00 00		mv s8, zero		x9/s1	0x00000000(0)
80000068	93 0C 00 00		mv s9, zero		x10/a0	0x00000000(0)
8000006C	13 0D 00 00		mv s10, zero		x11/a1	0x00000000(0)
80000070	93 0D 00 00		mv s11, zero		x12/a2	0x00000000(0)
80000074	13 0E 00 00		mv t3, zero		x13/a3	0x00000000(0)
80000078	93 0E 00 00		mv t4, zero		x14/a4	0x00000000(0)
8000007C	13 0F 00 00		mv t5, zero		x15/a5	0x00000000(0)
80000080	93 0F 00 00		mv t6, zero		x16/a6	0x00000000(0)
80000084	97 01 01 00		auipc gp, 0x10		x17/a7	0x00000000(0)
80000088	93 81 41 8A		addi gp, gp, -0x75c		x18/s2	0x00000000(0)
8000008C	17 81 01 00		auipc sp, 0x18		x19/s3	0x00000000(0)
80000090	13 01 81 8C		addi sp, sp, -0x738		x20/s4	0x00000000(0)
80000094	93 82 C1 80		addi t0, gp, -0x7f4		x21/s5	0x00000000(0)
80000098	17 83 01 00		auipc t1, 0x18		x22/s6	0x00000000(0)
8000009C	13 03 C3 88		addi t1, t1, -0x744		x23/s7	0x00000000(0)
--> 800000A0	23 A0 02 00		sw zero, 0(t0)		x24/s8	0x00000000(0)
800000A4	93 82 42 00		addi t0, t0, 4		x25/s9	0x00000000(0)
800000A8	E3 CC 62 FE		blt t0, t1, -8		x26/s10	0x00000000(0)
800000AC	73 25 A0 F1		csrr a0, mhartid		x27/s11	0x00000000(0)
Address			Data	Text	CSR	Value
					branchhit	0x00000498(1176)
					branchhith	0x00000000(0)
					branchmiss	0x00000013(19)
					branchmissl	0x00000000(0)
					branchnum	0x000004AB(1195)
					branchnumh	0x00000000(0)
					branchpred	0x000004AB(1195)
					branchpredl	0x00000000(0)
					charfifo	0x00000000(0)
					finish	0xFFFFFFFF(4294967295)
					marchid	0x19981001(429395969)
					mcause	0x00000000(0)
					mconfigptr	0x00000000(0)
					mcounteren	0x00000000(0)
					mcycle	0x000007D0(2000)
					mcycleh	0x00000000(0)
					mepc	0x00000000(0)
					mhartid	0x00000000(0)
					mie	0x00000000(0)
					mimpid	0x20220201(539099649)

send: b cycle 2000  
b: ok  
send: c  
c: ok



PipelineStatusWindow

Fetch1: PC = 0x80000A0 Jump | Fetch2: Busy Idle | Decode: Idle | Dispatch: IntBusy LSUBusy Busy InstWait: 1 StbufWait Stall | History: 99/99(Current) Prev(-) Next(+) Cycle: 2000

IntIssue: Busy ALU0: 0 ALU1: 0 BRU: 0 CSR: 0 DIV: 0 MUL0: 0 MUL1: 0 NextPortIndex: 0 Stall OutFeedback | LSUIssue: Busy LSUReadreg: Busy Stall

Commit: Idle Next: 41 EXCPC: 0x00000000 Flush JumpEN JumpPC: 0x00000000 I0: 40 I1: 0 I2: 0 I3: 0 | DIV: Busy: 6 LU\_L2: Stall Addr: 0x00000000

Fetch1->Fetch2:

0x80000A0:sw zero, 0(t0)  
0x80000A4:addi t0, t0, 4  
0x80000A8:blt t0, t1, -8  
<Empty>

Dispatch->Integer Issue:

11:0x80000A4:addi t0, t0, 4  
12:0x80000A8:blt t0, t1, -8  
14:0x80000A4:addi t0, t0, 4  
<Empty>

Integer Issue Queue:

0: <0,ready>5,0x80000A4:addi t0, t0, 4  
1: <0,ready>57,0x80000A4:addi t0, t0, 4  
2: <ready,ready>51,0x80000A4:addi t0, t0, 4  
3: <0,ready>58,0x80000A8:blt t0, t1, -8  
4: <0,ready>63,0x80000A4:addi t0, t0, 4  
5: <0,ready>0,0x80000A8:blt t0, t1, -8  
6: <0,ready>52,0x80000A8:blt t0, t1, -8  
7: <0,ready>54,0x80000A4:addi t0, t0, 4  
8: <0,ready>60,0x80000A4:addi t0, t0, 4  
9: <0,ready>6,0x80000A8:blt t0, t1, -8  
10: <0,ready>8,0x80000A4:addi t0, t0, 4  
11: <0,ready>55,0x80000A8:blt t0, t1, -8  
12: <0,ready>61,0x80000A8:blt t0, t1, -8  
13: <0,ready>2,0x80000A4:addi t0, t0, 4  
14: <0,ready>3,0x80000A8:blt t0, t1, -8  
15: <0,ready>9,0x80000A8:blt t0, t1, -8

Integer Issue->Integer Readreg:

49:0x80000A8:blt t0, t1, -8  
<Empty>

Integer Readreg->Execute:

ALU0: <Empty>  
ALU1: 48,0x80000A4:addi t0, t0, 4  
BRU: 46,0x80000A8:blt t0, t1, -8  
CSR: <Empty>  
DIV: <Empty>  
MUL0: <Empty>  
MUL1: <Empty>

LSU Issue->LSU Readreg:

<Empty>  
50:0x80000A0:sw zero, 0(t0)  
7,0x80000A0:<sw>sw zero, 0(t0)

LSU Readreg->Execute:

LU0: <Empty>  
SAU0: 47,0x80000A0:sw zero, 0(t0)  
SDU0: <Empty>

Execute->WB:

ALU0: 45,0x80000A4:addi t0, t0, 4  
ALU1: <Empty>  
BRU: <Empty>  
CSR: <Empty>  
DIV: <Empty>  
MUL0: <Empty>  
MUL1: <Empty>  
LU: <Empty>

Execute->Commit:

ALU0: 45,0x80000A4:addi t0, t0, 4  
ALU1: <Empty>  
BRU: <Empty>  
CSR: <Empty>  
DIV: <Empty>  
MUL0: <Empty>  
MUL1: <Empty>  
LU: <Empty>  
SAU: <Empty>  
SDU: <Empty>

Fetch2->Decode:

0x80000A4:addi t0, t0, 4  
0x80000A8:blt t0, t1, -8  
0x80000A0:sw zero, 0(t0)  
0x80000A4:addi t0, t0, 4  
0x80000A8:blt t0, t1, -8  
0x80000A0:sw zero, 0(t0)  
0x80000A4:addi t0, t0, 4  
0x80000A8:blt t0, t1, -8  
0x80000A0:sw zero, 0(t0)  
0x80000A4:addi t0, t0, 4  
0x80000A8:blt t0, t1, -8  
0x80000A0:sw zero, 0(t0)

Integer Issue Input:

<Empty>  
<Empty>  
<Empty>  
<Empty>

Dispatch->LSU Issue:

<Empty>  
<Empty>  
<Empty>  
<Empty>

LSU Issue Input:

<Empty>  
<Empty>  
<Empty>  
<Empty>

Decode->Rename:

0x80000A0:sw zero, 0(t0)  
0x80000A4:addi t0, t0, 4  
0x80000A8:blt t0, t1, -8  
0x80000A0:sw zero, 0(t0)  
0x80000A4:addi t0, t0, 4  
0x80000A8:blt t0, t1, -8

ROB:

41:41,0x80000A0:sw zero, 0(t0)(Finished)  
42:42,0x80000A4:addi t0, t0, 4(Finished)  
43:43,0x80000A8:blt t0, t1, -8(Finished)  
44:44,0x80000A0:sw zero, 0(t0)(Finished)  
45:45,0x80000A4:addi t0, t0, 4(Unfinish)  
46:46,0x80000A8:blt t0, t1, -8(Unfinish)  
47:47,0x80000A0:sw zero, 0(t0)(Unfinish)  
48:48,0x80000A4:addi t0, t0, 4(Unfinish)  
49:49,0x80000A8:blt t0, t1, -8(Unfinish)  
50:50,0x80000A0:sw zero, 0(t0)(Unfinish)  
51:51,0x80000A4:addi t0, t0, 4(Unfinish)  
52:52,0x80000A8:blt t0, t1, -8(Unfinish)  
53:53,0x80000A0:sw zero, 0(t0)(Unfinish)  
54:54,0x80000A4:addi t0, t0, 4(Unfinish)  
55:55,0x80000A8:blt t0, t1, -8(Unfinish)  
56:56,0x80000A0:sw zero, 0(t0)(Unfinish)  
57:57,0x80000A4:addi t0, t0, 4(Unfinish)  
58:58,0x80000A8:blt t0, t1, -8(Unfinish)  
59:59,0x80000A0:sw zero, 0(t0)(Unfinish)  
60:60,0x80000A4:addi t0, t0, 4(Unfinish)

Rename->Dispatch:

15,0x80000A8:blt t0, t1, -8  
16,0x80000A0:sw zero, 0(t0)  
17,0x80000A4:addi t0, t0, 4  
18,0x80000A8:blt t0, t1, -8

RAT:

0: x5<invalid>, x5<invalid>, 0x00000000<invalid>  
1: x5<invalid>, x5<invalid>, 0x00000000<invalid>  
2: x5<invalid>, x5<invalid>, 0x00000000<invalid>  
3: x5<invalid>, x5<invalid>, 0x00000000<invalid>  
4: x5<invalid>, x5<invalid>, 0x00000000<invalid>  
5: x5<invalid>, x5<invalid>, 0x00000000<invalid>  
6: x5<invalid>, x5<invalid>, 0x00000000<invalid>  
7: x5<invalid>, x5<invalid>, 0x00000000<invalid>  
8: x5<invalid>, x5<invalid>, 0x00000000<invalid>  
9: x5<invalid>, x5<invalid>, 0x00000000<invalid>  
10: x5<invalid>, x5<invalid>, 0x00000000<invalid>

enable : True  
value : 0xFE62CCE3(4267887843)  
valid : 0x00000001(1)  
last\_uop : True  
rob\_id : 0x00000037(55)  
pc : 0x80000A8(2147483816)  
imm : 0xFFFFFFFF(4294967288)  
has\_exception : False  
exception\_id : instruction\_access\_fault  
exception\_value : 0x80000A8(2147483816)  
rs1 : 0x00000005(5)  
arg1\_src : reg  
rs1\_need\_map : True  
rs1\_phy : 0x0000005C(92)  
rs2 : 0x00000006(6)  
arg2\_src : reg  
rs2\_need\_map : True  
rs2\_phy : 0x00000044(68)  
rd : 0x00000000(0)  
rd\_enable : False  
need\_rename : False  
rd\_phy : 0x00000000(0)  
csr : 0x00000000(0)  
op : blt  
op\_unit : bru  
sub\_op : blt  
Instruction : blt t0, t1, -8