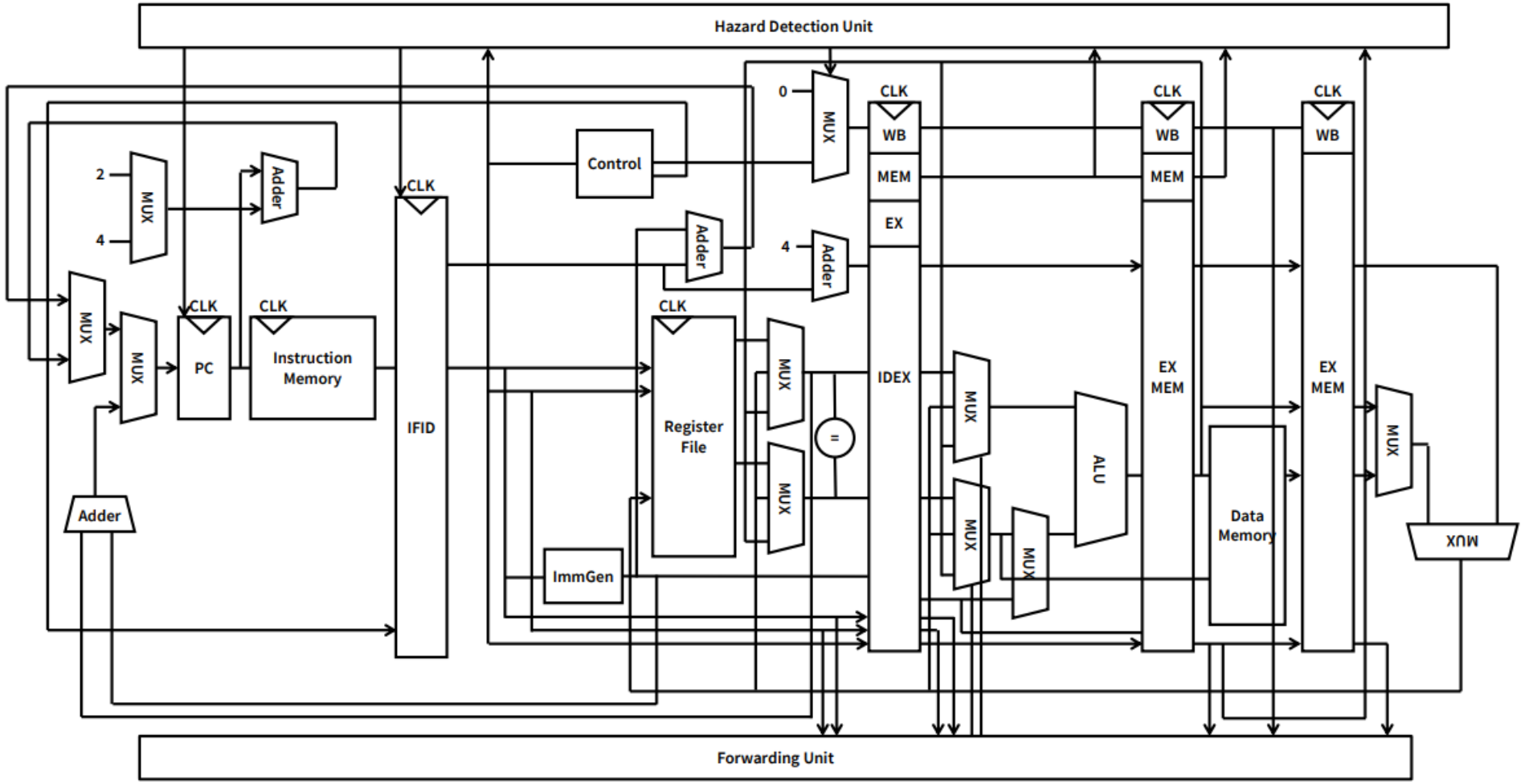
DSD Final Report

Baseline

- Overall Architecture
- Jump-related instructions
- Branch
- Trade offs

Overall Architecture



Jump-related instructions – jal/jalr

- IF, ID or EX
  - IF stage：No cycle waste
  - ID stage ： Imm Gen share
  - EX stage ： ALU share, forward Resolved

- jalr Target
  - ID forwarding ：2 level deep datapath + load use hazard (2 cycle stall)
  - EX will be better choice

Branch

- ID Branching
  - Textbook method ： lack of consider ID forwarding
  - ID forwarding ： more hazard

- EX Branching
  - 1 more cycle cost
  - Better data dependency
  - Benefits from good branch predictions

Hazard

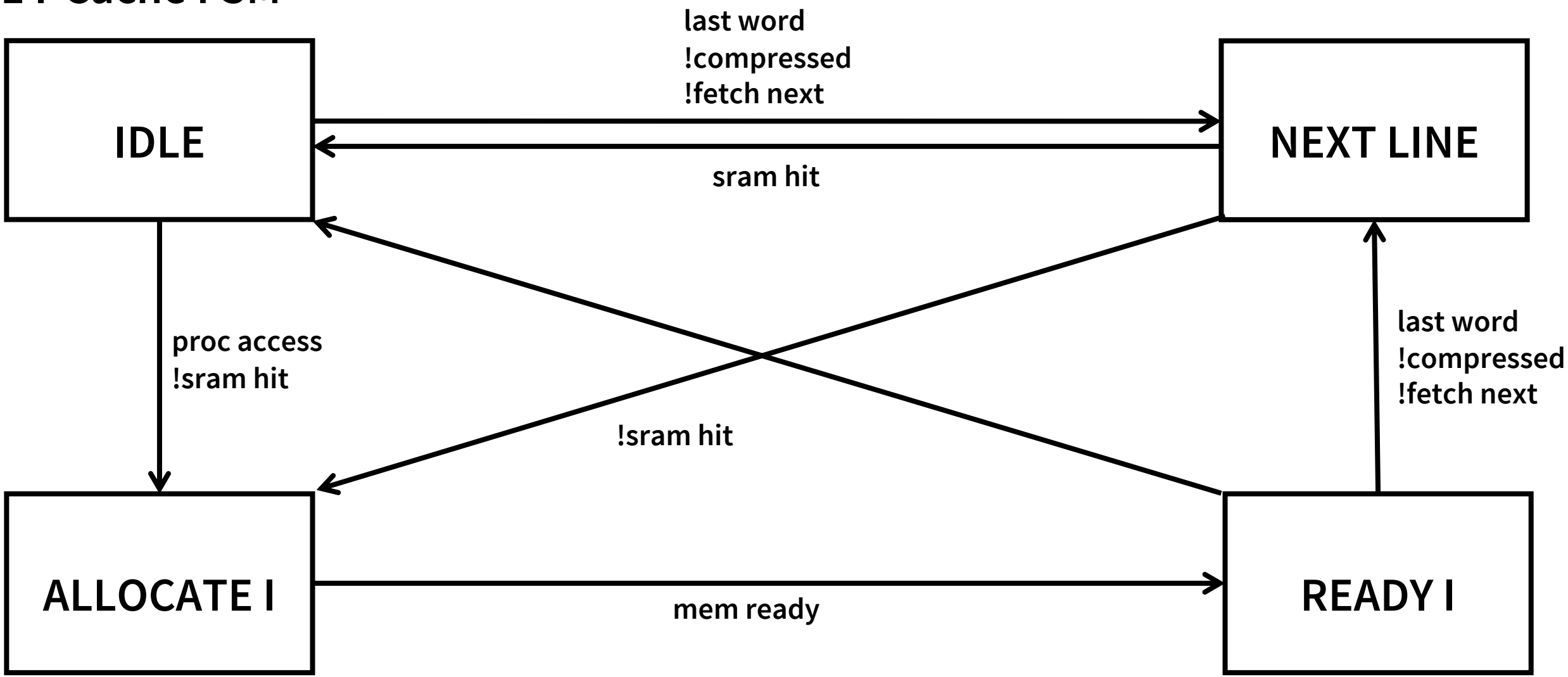
- Forwarding
  - WB MEM to EX
- Load-Use
  - Insert 1 Bubble
- Control
  - Flush at branch miss, jal and jalr

Cache

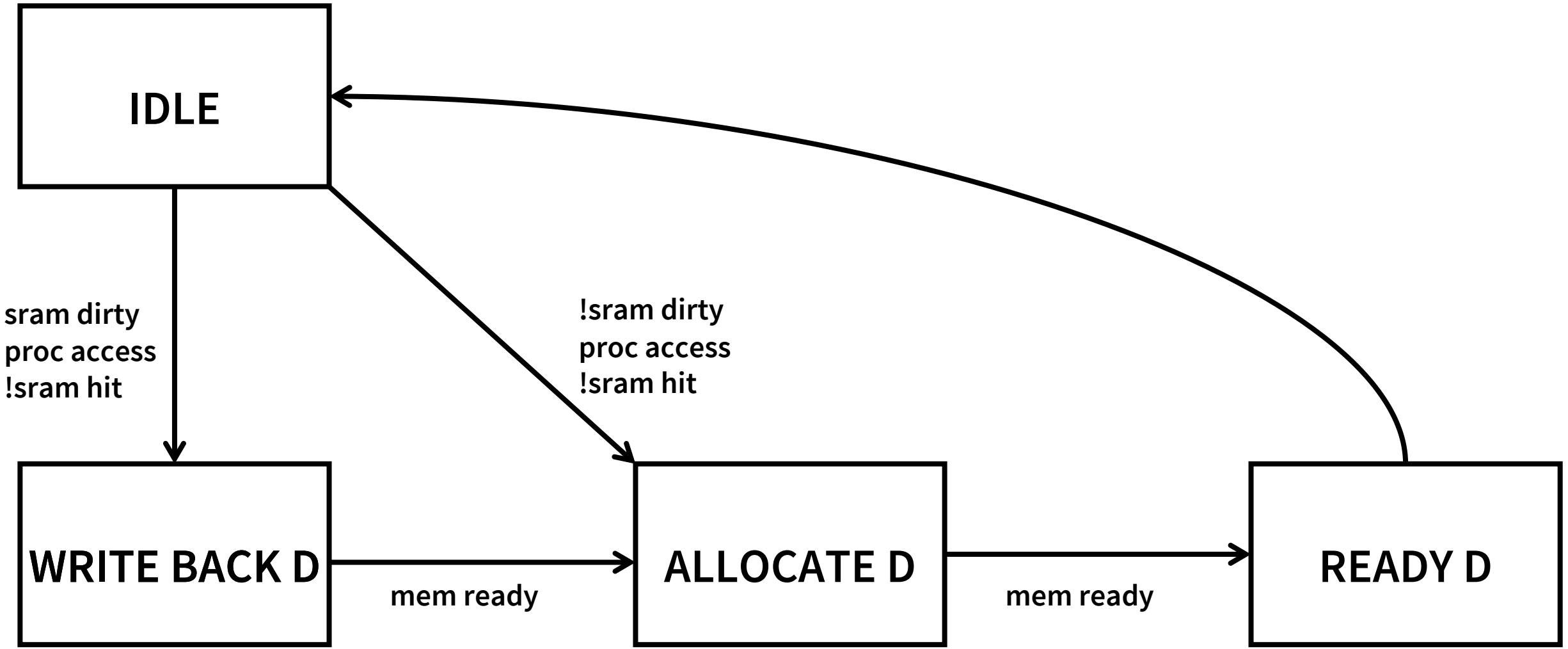
Spec

- L1 Cache ： 32 words + 2-way associative
  - I-Cache ： Read Only
  - D-Cache ： Both
  - LRU Replacement
  - Write Back
- L2 Cache ： 128 words + 2-way associative
  - Unified
  - LRU Replacement
  - Write Back

L1 I-Cache FSM



L1 D-Cache FSM



L2 Cache

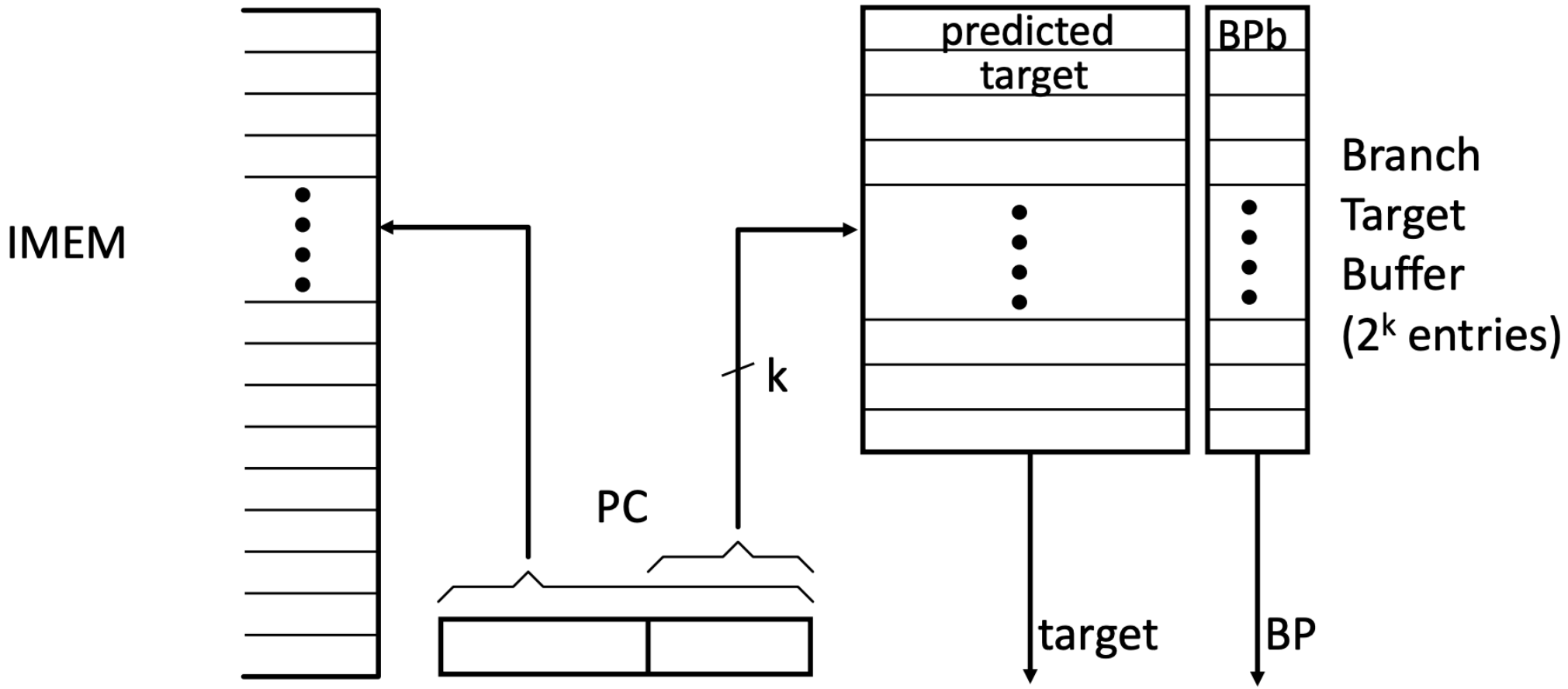
- Unified
  - Use 1 bit to store I or D
  - Save area
- FSM
  - 2 FSM to control I + D
- Collision
  - Arbiter gives permission sequentially

Performace

- No L2 cache ： 297377500 PS
- With L2 cache ： 232622500 PS
- 22% Improvement!!!

Branch Prediction

Idea

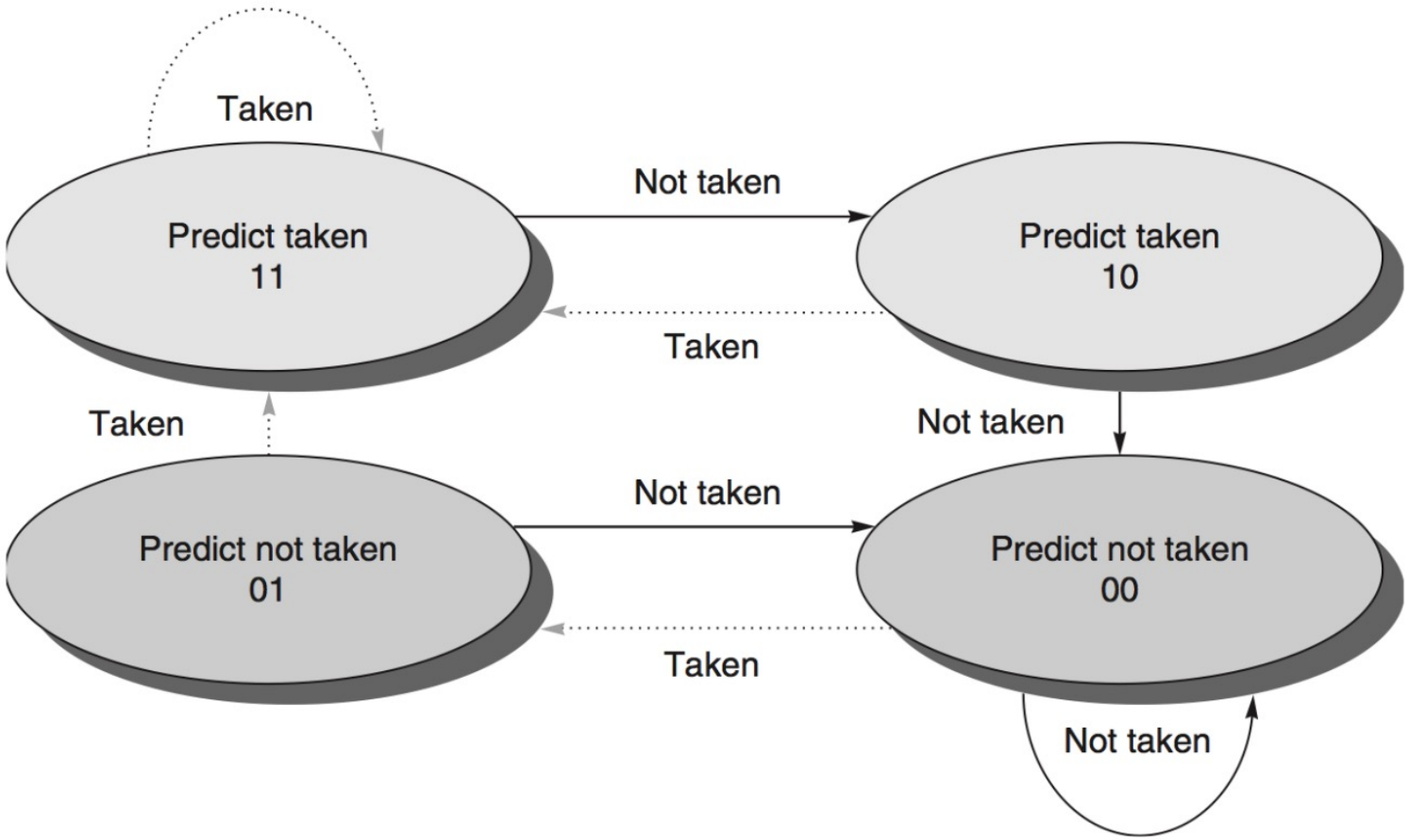

IMEM
PC
k
predicted
target
BPb
Branch
Target
Buffer
(2^k entries)
target
BP

Branch History Table?

- Can calculate Target at IF
  - Cause 2 adder delay(+ 4, + Imm)
  - Save area

- Tradeoff between area and time

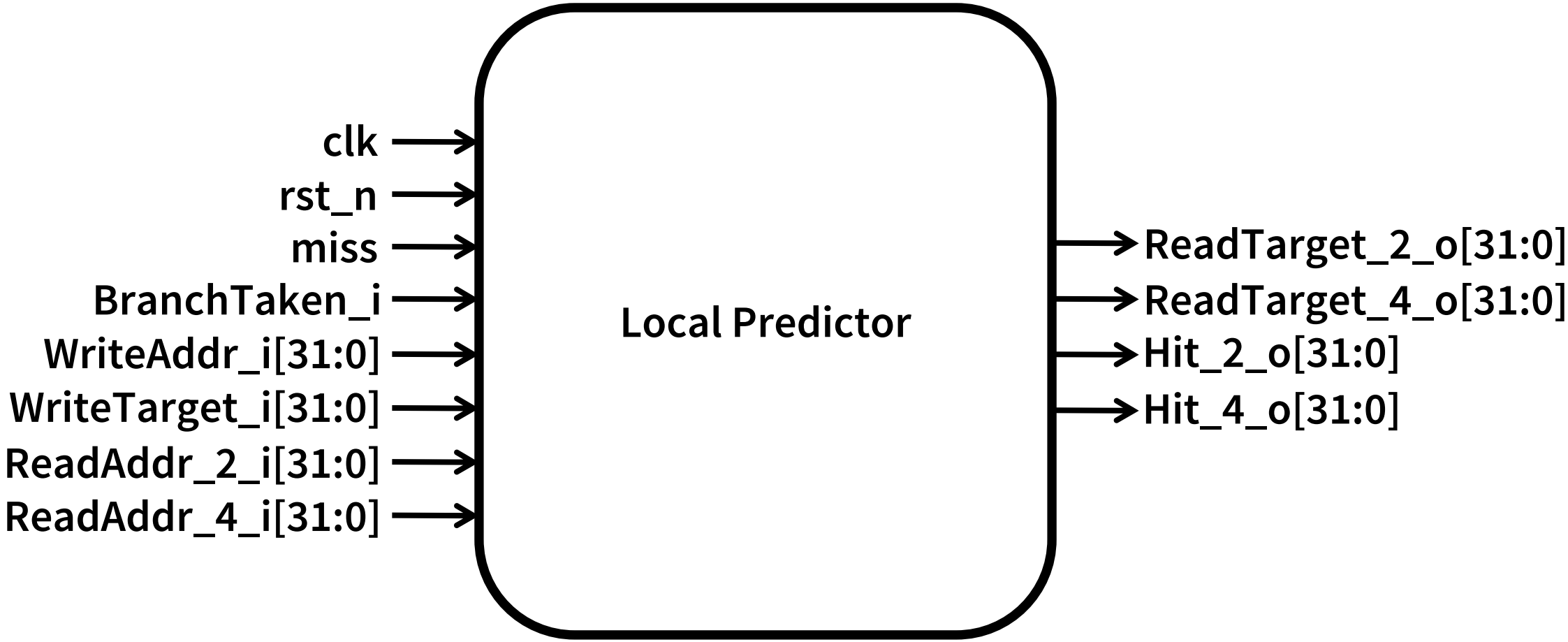
FSM 2-bit


Taken
Predict taken
11
Not taken
Predict taken
10
Taken
Taken
Not taken
Predict not taken
01
Not taken
Predict not taken
00
Taken
Not taken

Collisions

- Store Tag bits to double check

Local Predictor



利用Local predictor建立Branch History Table，做出判斷，減少Global Predictor判斷錯誤的情況

Performance

- With BP ：890750
- No BP ： 1216250
- 26.7% Improvement

C Extension

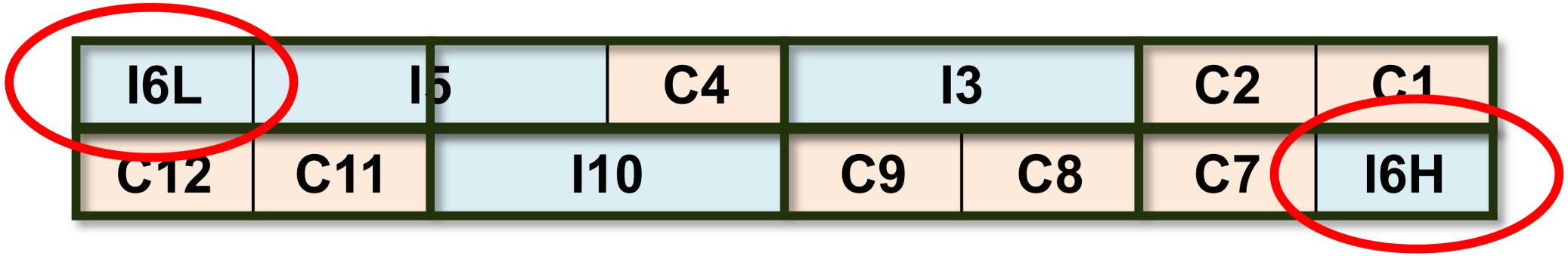
Compressed Instructions

- Decompressor:
  - Translate 16/32 bit Instruction to 32
  - Fully Combinational

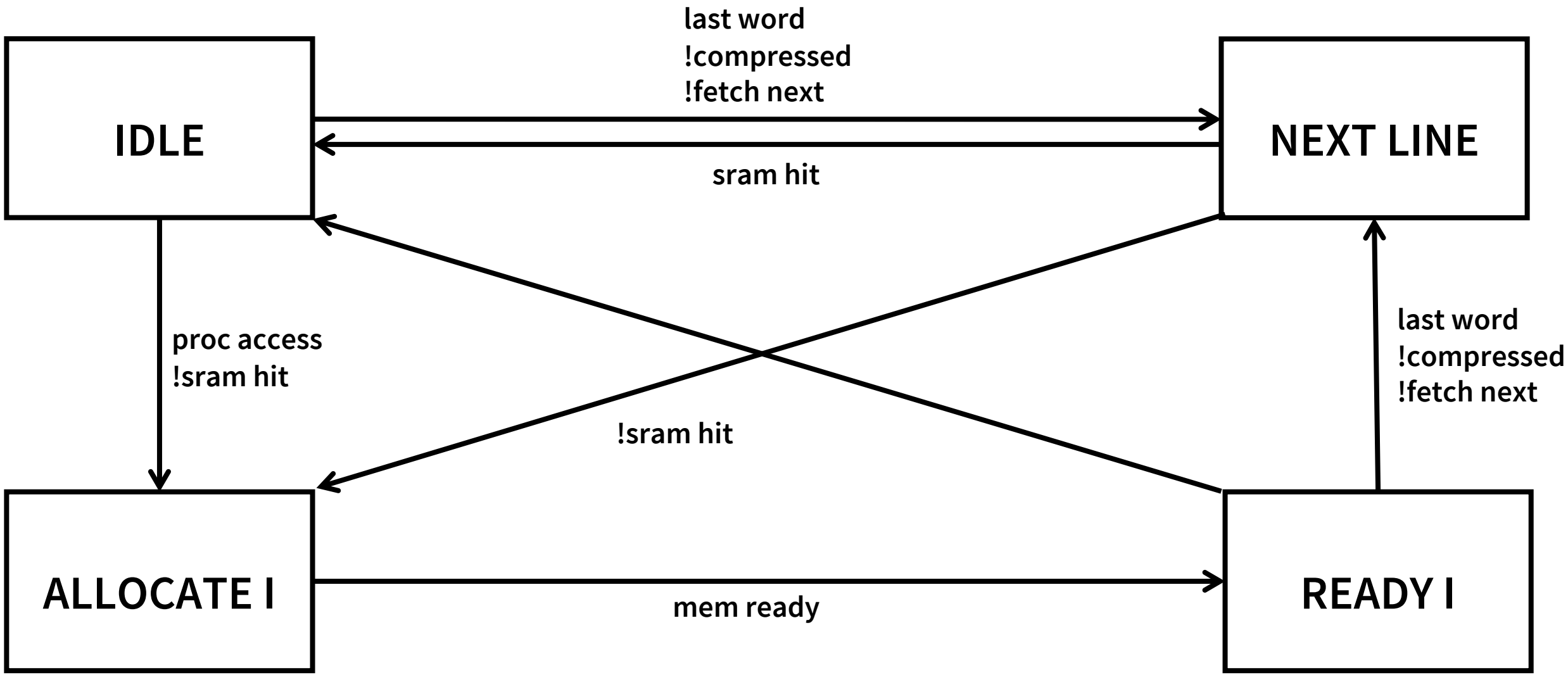
- PC modify:
  - PC += 2 or 4
  - Tag bit

IF Issues

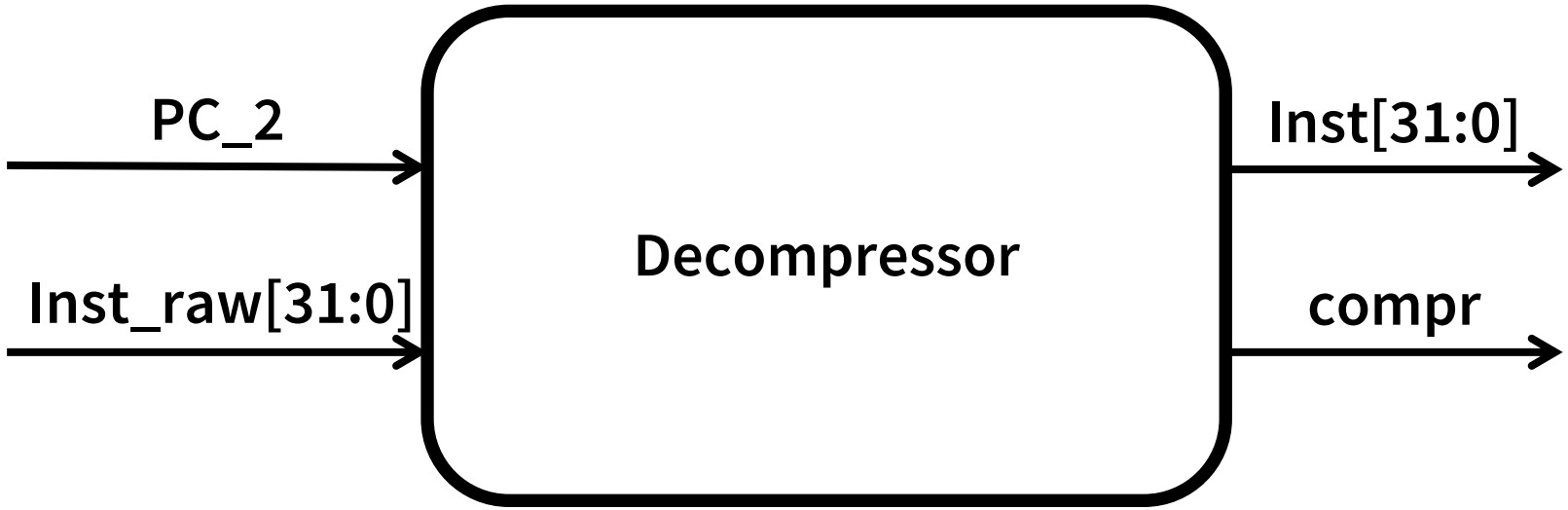
- 2 offset don’t fit to original cache
  - Modify cache to support half Word Read
- 32 bit Instruction may cross block
  - Modify FSM to buffer the first half and merge with second half



Buffer I-Cache FSM



Decompressor



Performance

- Uncompressed: 550498500
- Compressed.　: 457042500
- 17% Improvement

Merging Extensions

Difficulties

- L2 Cache
  - Large area, long datapath at decoder
  - I really don’t know how to improve
- Branch Prediction
  - Big Mux after BHT
  - Solution ： Parallel output both PC+2, PC+4 result and MUX out at the end
- Decompressor
  - Critical path for whole IF
  - Solution ： Select few bit out to generate simple signals (branch, jump…)

Tradeoffs

IF stage

- jal at ID (share Imm, hide delay to ID)
- jalr at EX (complete forwarding, less hazard)
- BHT is used (remove all adder at IF)
- Use MUX to hide latency of late signal (Decompressor)

L2 Cache

- Huge area (about 600K um^2) with 20% speedup
- Not seems worthy to use at AT score

Score

$$A = 317261.033111 \ (um^2)$$
$$T = 457042.500 \ (ns)$$
$$AT = 1.45E11$$

On QuickSort n=256 testbench

What I’ve learned

Appendix

Appendix

- Decompressor

Port name	Direction	Type
PC_2	input	
inst_raw	input	[31:0]
inst	output	[31:0]
compr	output	

- Adder

Port name	Direction	Type
data1_in	input	[31:0]
data2_in	input	[31:0]
data_o	output	[31:0]

Appendix

- ALU_Control

Port name	Direction	Type
funct3_i	input	[2:0]
funct7_i	input	
ALUOp_i	input	[1:0]
ALUCtrl_o	output	[3:0]

- ALU

Port name	Direction	Type
data1_i	input	signed [31:0]
data2_i	input	signed [31:0]
ALUCtrl_i	input	[3:0]
data_o	output	[31:0]
Zero_o	output	

Appendix

- Compare

Port name	Direction	Type
equal_i	input	
func3_0_i	input	
branch_i	input	
branch_taken_o	output	

- Imm_Gen

Port name	Direction	Type
instruction_i	input	[31:0]
imm_o	output	[31:0]

Appendix

- Control

Port name	Direction	Type
Op_i	input	[6:0]
NoOp_i	input	
jalr_o	output	
jal_o	output	
branch_o	output	
MemRead_o	output	
MemtoReg_o	output	
ALUOp_o	output	[1:0]
MemWrite_o	output	
ALUSrc_o	output	
RegWrite_o	output	

Appendix

- EX_MEM

Port name	Direction	Type
clk	input	
rst_n	input	
Stall_i	input	
flush_i	input	
ctrl_i	input	[3: 0]
ctrl_o	output	[3: 0]
ALUResult_i	input	signed [31: 0]
ALUResult_o	output	[31: 0]
RS2data_i	input	signed [31: 0]
RS2data_o	output	[31: 0]
RDaddr_i	input	[4: 0]
RDaddr_o	output	[4: 0]

Appendix

- **Forwarding_Unit**

Port name	Direction	Type
RS1_i	input	[4:0]
RS2_i	input	[4:0]
MEM_RD_i	input	[4:0]
WB_RD_i	input	[4:0]
MEM_RegWrite_i	input	
WB_RegWrite_i	input	
FowardA_o	output	[1:0]
FowardB_o	output	[1:0]

Appendix

- Hazard_Detection

Port name	Direction	Type
ID_RS1_i	input	[4:0]
ID_RS2_i	input	[4:0]
EX_RDaddr_i	input	[4:0]
EX_MEMRead_i	input	
NoOP_o	output	
PCWrite_o	output	
Stall_o	output	

Appendix

- PC

Port name	Direction	Type
clk	input	
rst_n	input	
stall_i	input	
PCWrite_i	input	
pc_i	input	[31:0]
pc_o	output	[31:0]

Appendix

- cache_sram_2way

Port name	Direction	Type
clk	input	
rst	input	
addr_i	input	[29:0]
wdata_i	input	[155:0]
write_i	input	
rdata_o	output	[155:0]
hit_o	output	

Appendix

- cache_sram_l2

Port name	Direction	Type
clk	input	
rst	input	
addr_i	input	[27:0]
wdata_i	input	[153:0]
write_i	input	
I_D	input	
rdata_o	output	[153:0]
hit_o	output	

Appendix

- Registers

Port name	Direction	Type
clk	input	
rst_n	input	
RS1addr_i	input	[4:0]
RS2addr_i	input	[4:0]
RDaddr_i	input	[4:0]
RDdata_i	input	signed [31:0]
RegWrite_i	input	
RS1data_o	output	signed [31:0]
RS2data_o	output	signed [31:0]

Appendix

- ID_EX

Port name	Direction	Type	Port name	Direction	Type	Port name	Direction	Type
clk	input		RS2data_i	input	[31: 0]	func3_0_i	input	
rst_n	input		RS2data_o	output	[31: 0]	func3_0_o	output	
Stall_i	input		jump_i	input		BP_hit_i	input	
flush_i	input		jump_o	output		BP_hit_o	output	
ctrl_i	input	[6: 0]	jalr_i	input		pc_imm_i	input	[31: 0]
ctrl_o	output	[6: 0]	jalr_o	output		pc_imm_o	output	[31: 0]
RS1data_i	input	[31: 0]	branch_i	input		pc_plus_i	input	[31: 0]
RS1data_o	output	[31: 0]	branch_o	output		pc_plus_o	output	[31: 0]

Appendix

- D_cache/I_cache

Port name	Direction	Type	Port name	Direction	Type
clk	input		mem_read_o	output	
proc_reset_i	input		mem_write_o	output	
proc_read_i	input		mem_addr_o	output	[27:0]
proc_write_i	input		mem_rdata_i	input	[127:0]
proc_addr_i	input	[29:0]	mem_wdata_o	output	[127:0]
proc_rdata_o	output	[31:0]	mem_ready_i	input	
proc_wdata_i	input	[31:0]			
proc_stall_o	output				

Appendix

- ID_EX

Port name	Direction	Type	Port name	Direction	Type	Port name	Direction	Type
imm_i	input	[31: 0]	RDaddr_i	input	[4: 0]			
imm_o	output	[31: 0]	RDaddr_o	output	[4: 0]			
funct_i	input	[9: 0]						
funct_o	output	[9: 0]						
RS1addr_i	input	[4: 0]						
RS1addr_o	output	[4: 0]						
RS2addr_i	input	[4: 0]						
RS2addr_o	output	[4: 0]						

Appendix

- L2cache

Port name	Direction	Type	Port name	Direction	Type	Port name	Direction	Type
clk	input		memi_read	output		L1d_addr_i	input	[27:0]
proc_reset	input		memi_write	output		L1d_wdata_i	input	[127:0]
L1i_read_i	input		memi_addr	output	[27:0]	L1d_rdata_o	output	[127:0]
L1i_write_i	input		memi_rdata	input	[127:0]	L1d_ready	output	
L1i_addr_i	input	[27:0]	memi_wdata	output	[127:0]	memd_read	output	
L1i_wdata_i	input	[127:0]	memi_ready	input		memd_write	output	
L1i_rdata_o	output	[127:0]	L1d_read_i	input		memd_addr	output	[27:0]
L1i_ready	output		L1d_write_i	input		memd_rdata	input	[127:0]
memd_wdata	output	[127:0]	memd_ready	input				

Appendix

- IF_ID

Port name	Direction	Type	Port name	Direction	Type	Port name	Direction	Type
clk	input		pc_o	output	[31: 0]			
rst_n	input		pc_imm_i	input	[31: 0]			
Stall_i	input		pc_imm_o	output	[31: 0]			
instr_i	input	[31: 0]	BP_hit_i	input				
instr_o	output	[31: 0]	BP_hit_o	output				
pc_plus_i	input	[31: 0]						
pc_plus_o	output	[31: 0]						
pc_i	input	[31: 0]						

Appendix

- Prediction

Port name	Direction	Type	Port name	Direction	Type
clk	input		ReadAddr_4_i	input	[31:0]
rst_n	input		ReadTarget_4_o	output	[31:0]
miss	input				
BranchTaken_i	input		Hit_2_o	output	
WriteAddr_i	input	[31:0]	Hit_4_o	output	
WriteTarget_i	input	[31:0]			
ReadAddr_2_i	input	[31:0]			
ReadTarget_2_o	output	[31:0]			

Appendix

- PC_Control

Port name	Direction	Type	Port name	Direction	Type	Port name	Direction	Type
imm_ext	input	[31:0]	IF_jalr_i	input		PC_plus_o	output	[31:0]
PC_i	input	[31:0]	IF_jal_i	input		PC_imm_o	output	[31:0]
PC_branch	input	[31:0]	branch_pred	input				
PC_branch_target	input	[31:0]	compressed	input				
			miss	input				
PC_jalr	input	[31:0]	PC_o	output	[31:0]			
PC_jal	input	[31:0]	PC_plus2_o	output	[31:0]			
jal_i	input		PC_plus4_o	output	[31:0]			
jalr_i	input							

Appendix

- MEM_WB

Port name	Direction	Type	Port name	Direction	Type
clk	input		MemData_o	output	[31: 0]
rst_n	input		RDaddr_i	input	[4: 0]
Stall_i	input		RDaddr_o	output	[4: 0]
ctrl_i	input	[1: 0]			
ctrl_o	output	[1: 0]			
ALUResult_i	input	signed [31: 0]			
ALUResult_o	output	[31: 0]			
MemData_i	input	signed [31: 0]			

Appendix

- Jump_Imm_Gen

Port name	Direction	Type
instruction_i	input	[31:0]
imm_o	output	[31:0]

- MUX4

Port name	Direction	Type
data00_i	input	[31:0]
data01_i	input	[31:0]
data10_i	input	[31:0]
data11_i	input	[31:0]
select_i	input	[1:0]
data_o	output	[31:0]