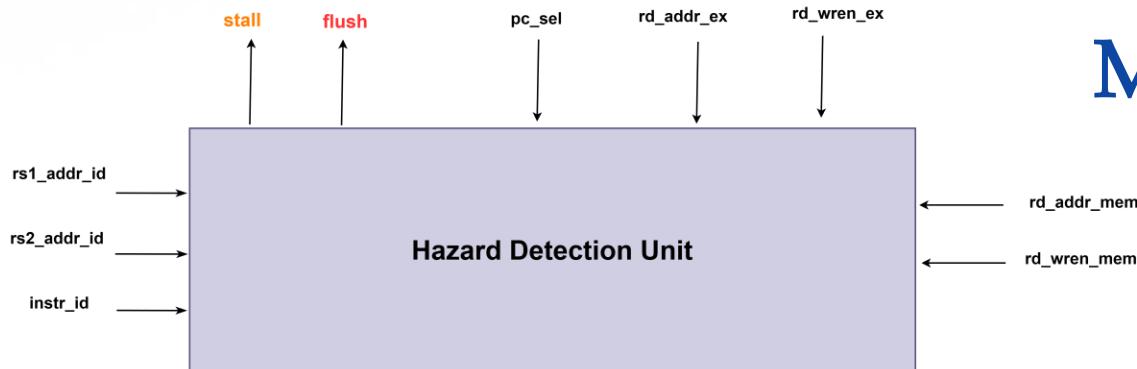
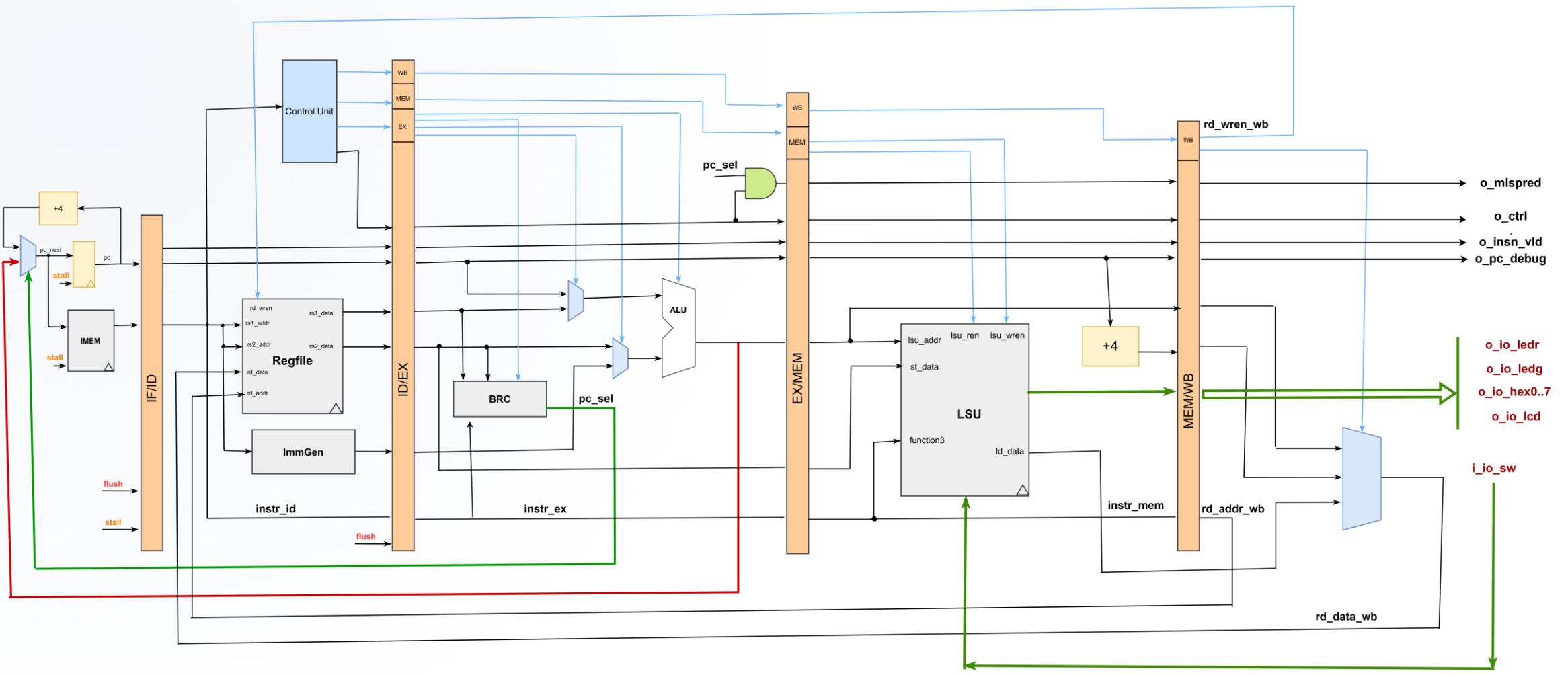




BÁO CÁO MILESTONE 3

THIẾT KẾ PIPELINED PROCESSOR RV32I

EE3203 – KIẾN TRÚC MÁY TÍNH | Lớp L02 | Nhóm 7



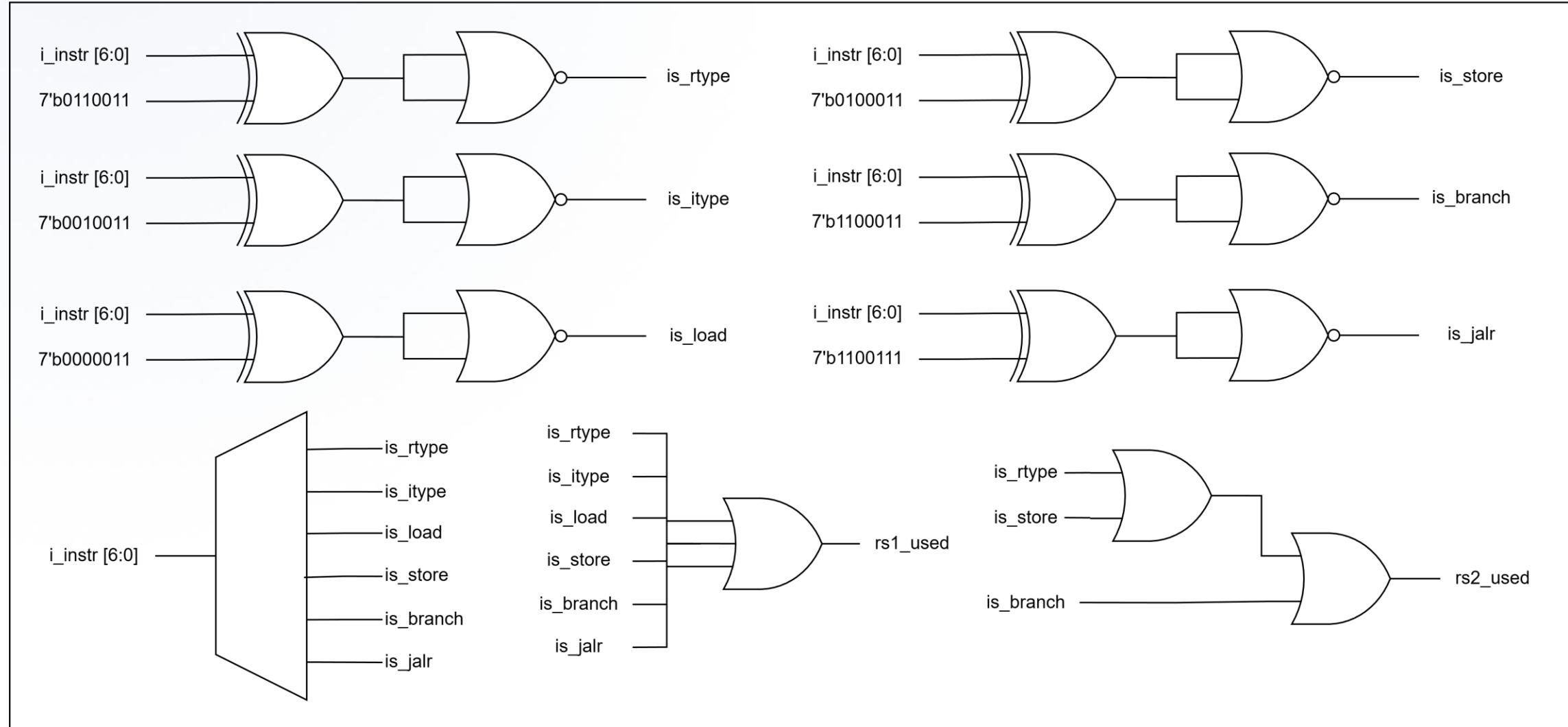
Model 1 : Non-Forwarding

Tổng quan Datapath

Model 1 : Non-Forwarding

Hazard Detection Unit

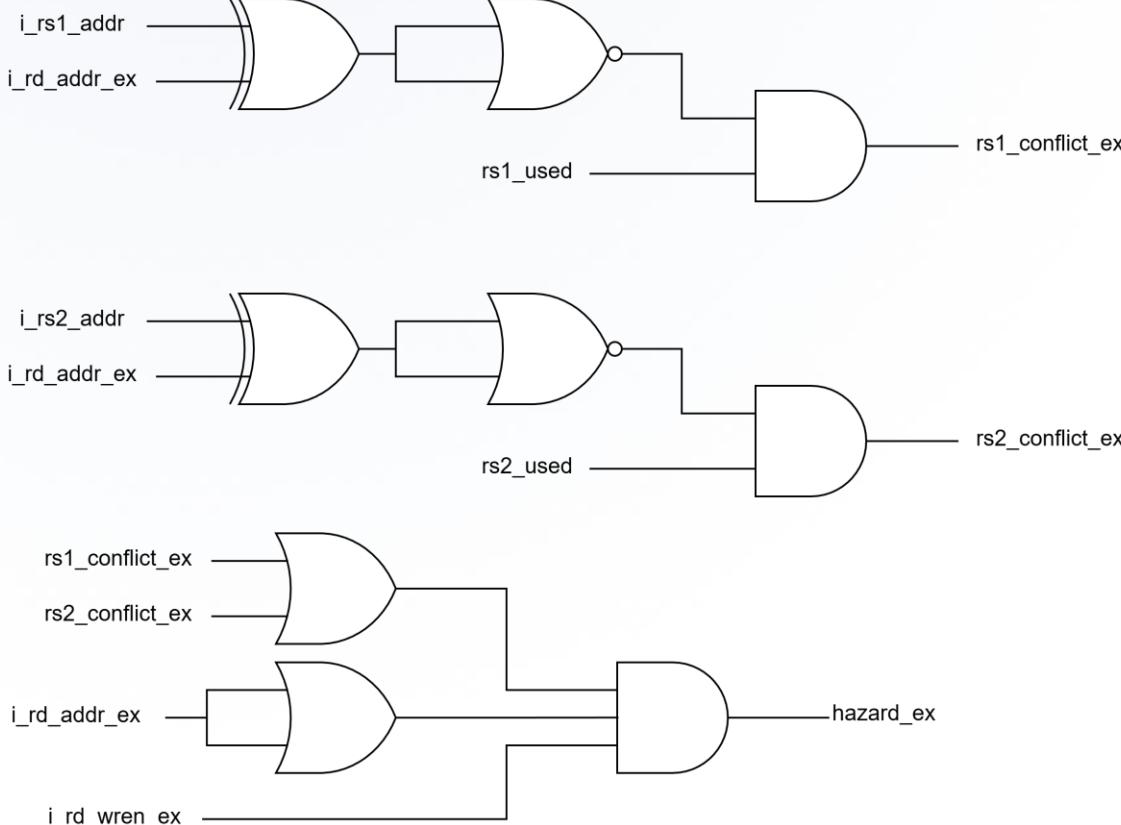
Instruction Decoder



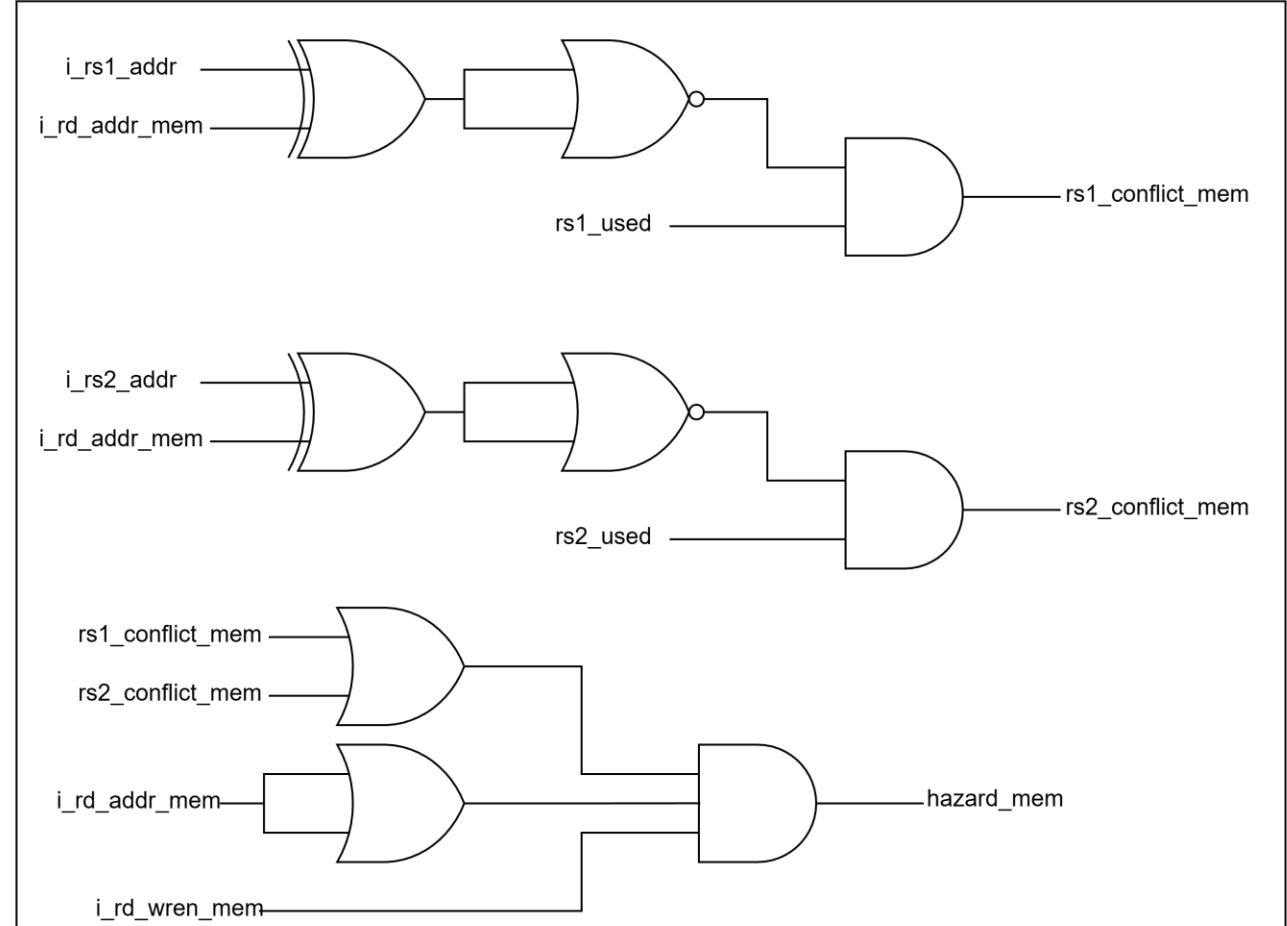
Model 1 : Non-Forwarding

Hazard Detection Unit

Data Hazard EX Stage

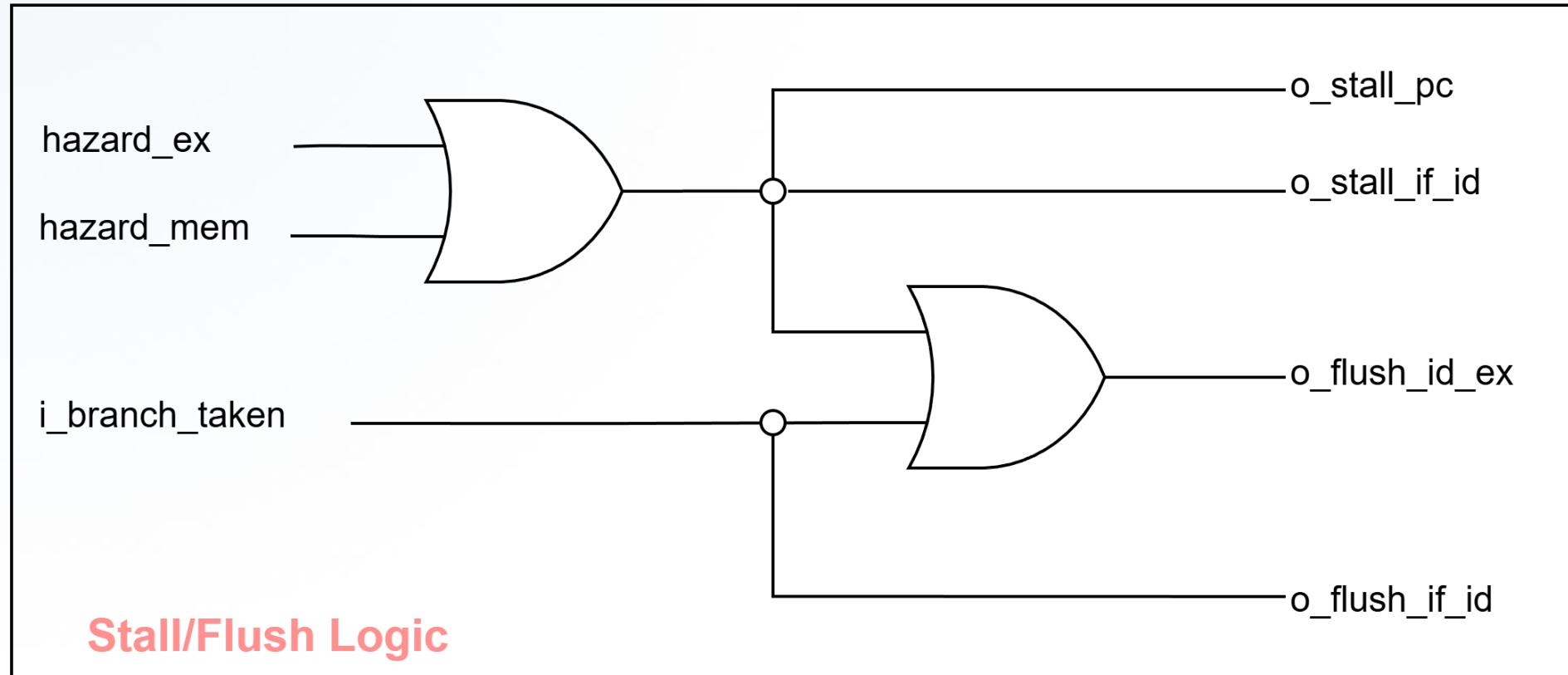


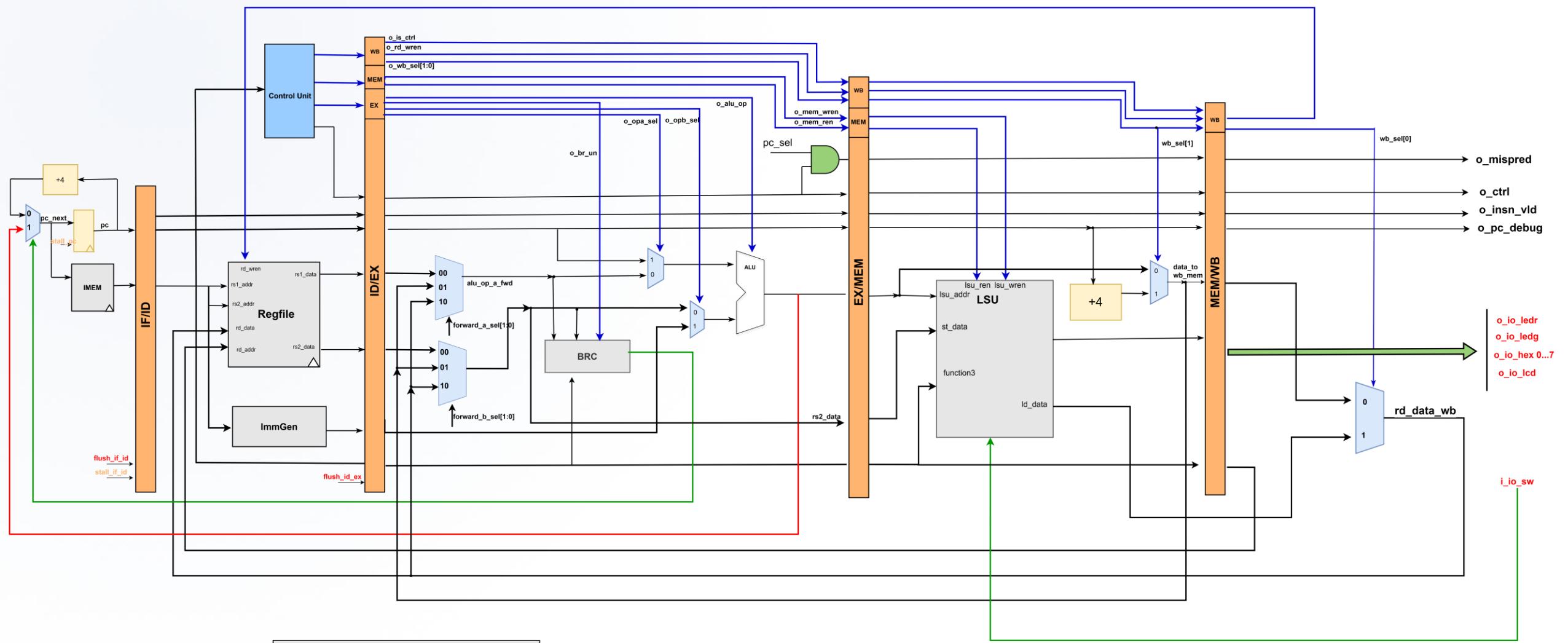
Data Hazard MEM Stage



Model 1 : Non-Forwarding

Hazard Detection Unit

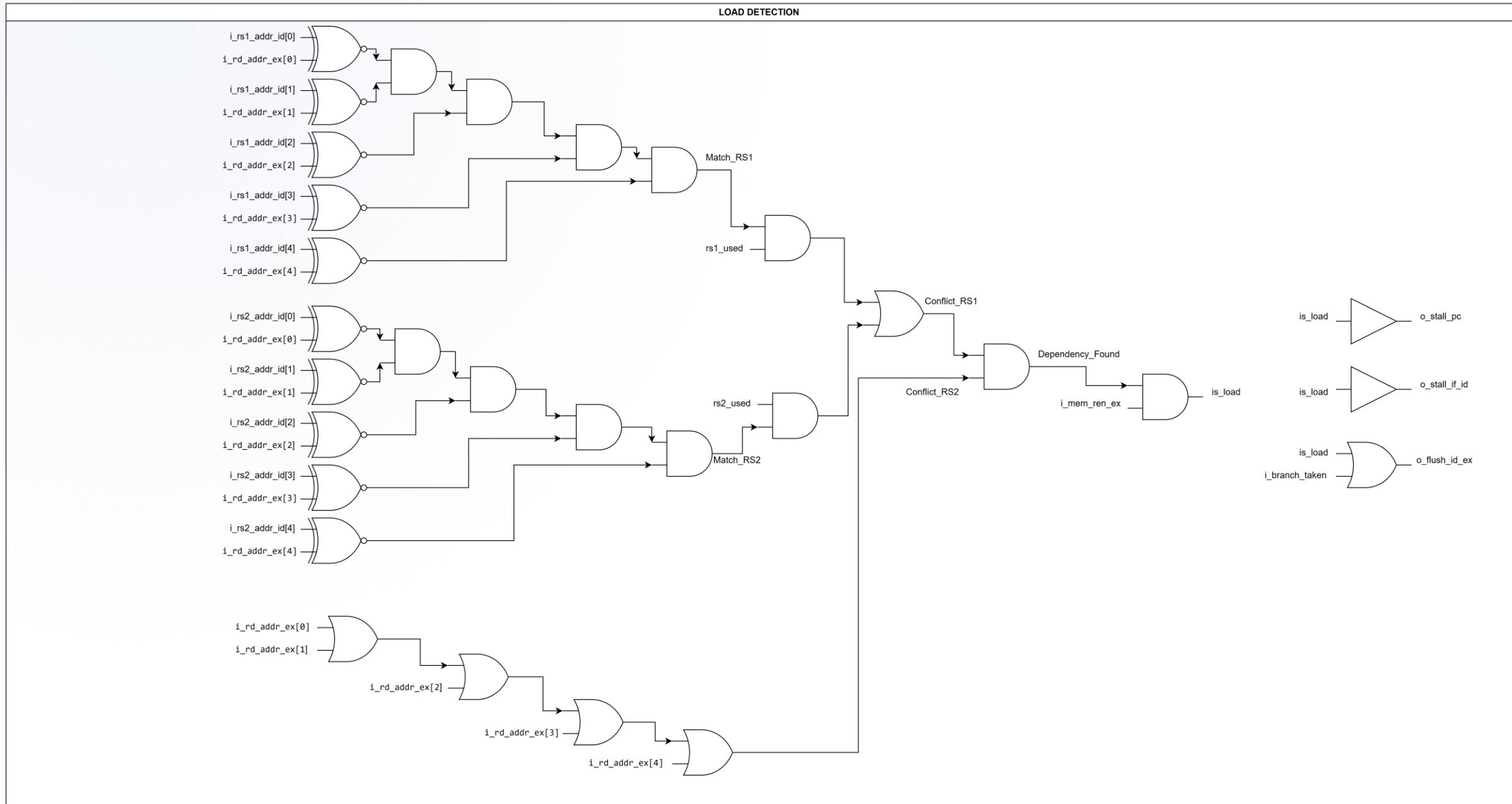




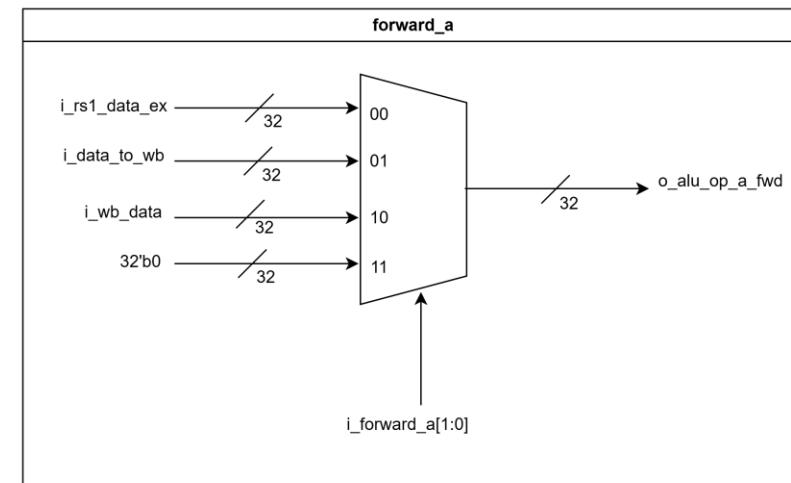
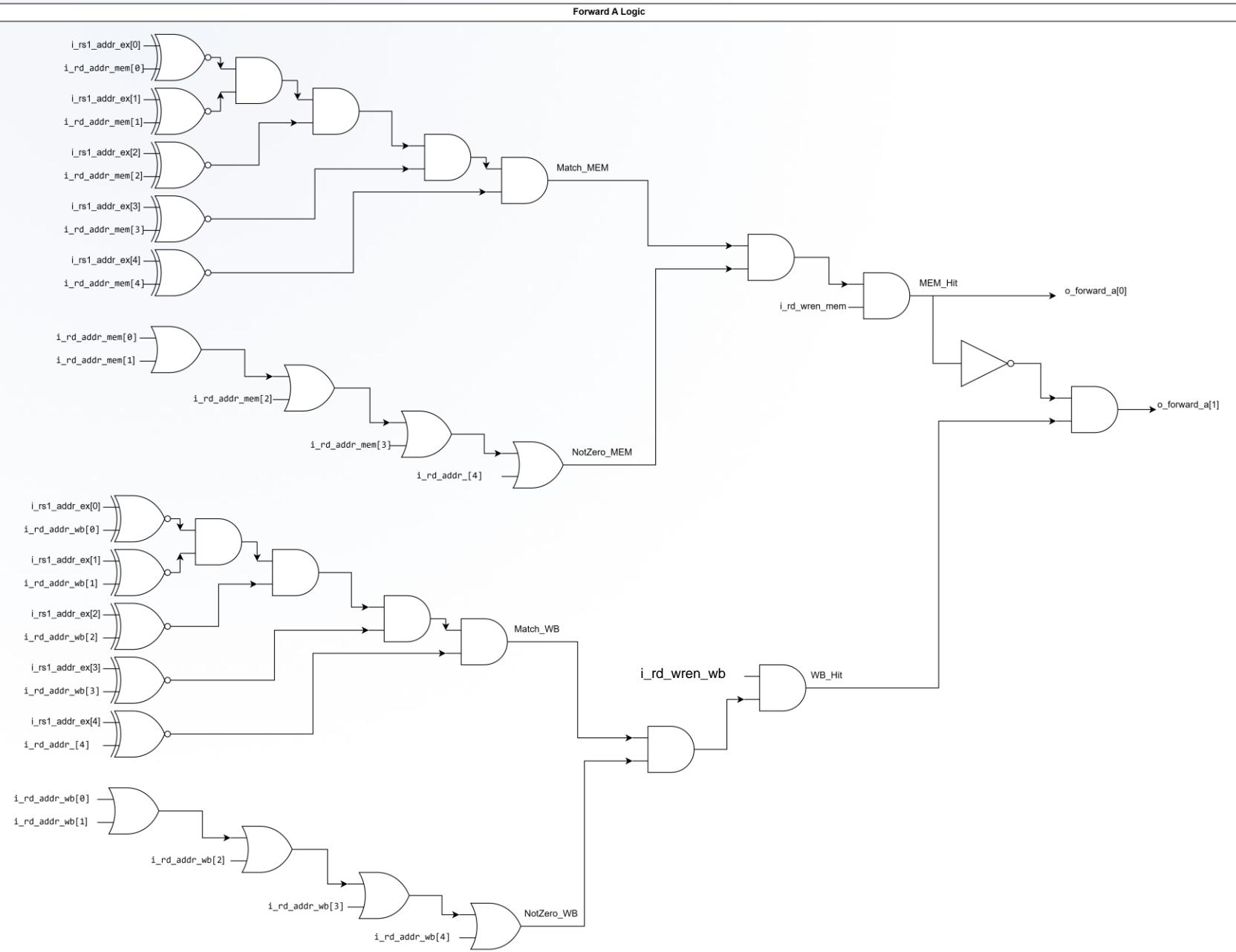
Model 2: Forwarding

Tổng quan Datapath

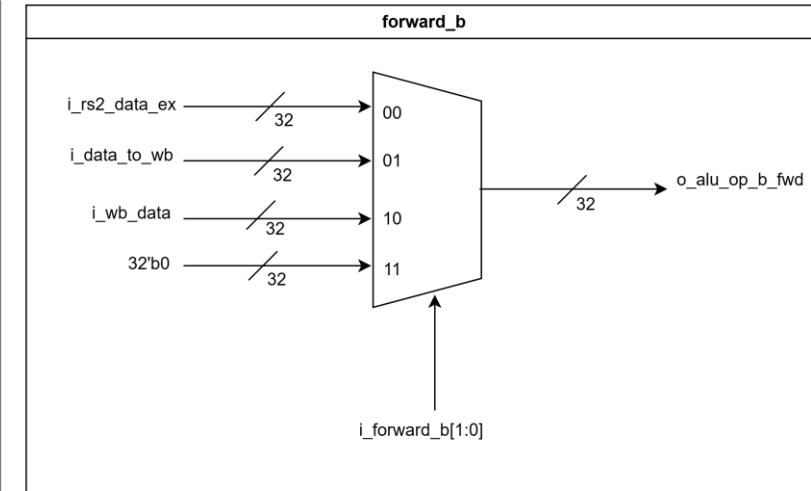
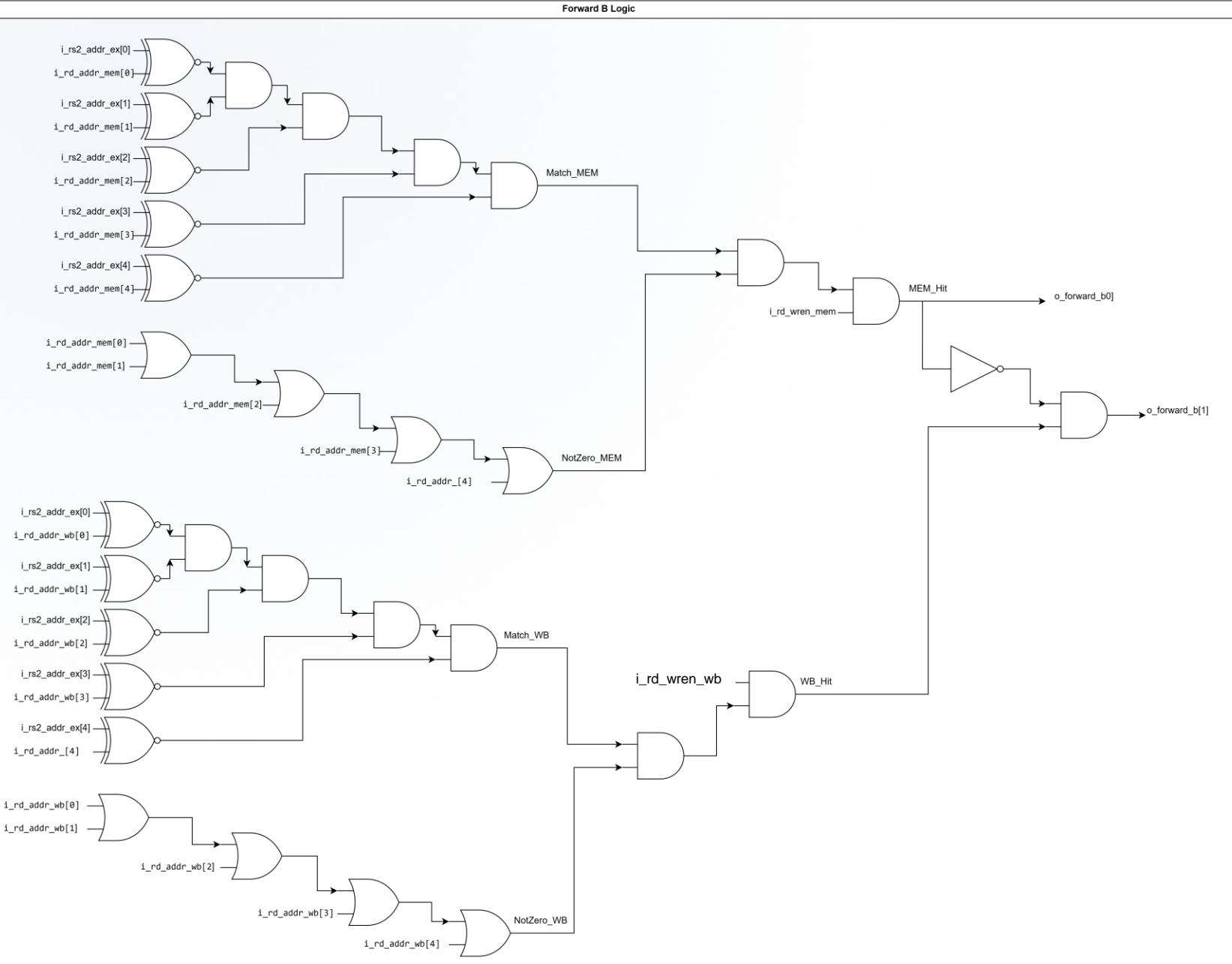
Model 2: Forwarding – Hazard Detection Forwarding Unit



Model 2: Forwarding – Hazard Detection Forwarding Unit



Model 2: Forwarding – Hazard Detection Forwarding Unit



Khối ALU (Arithmetic Logic Unit)

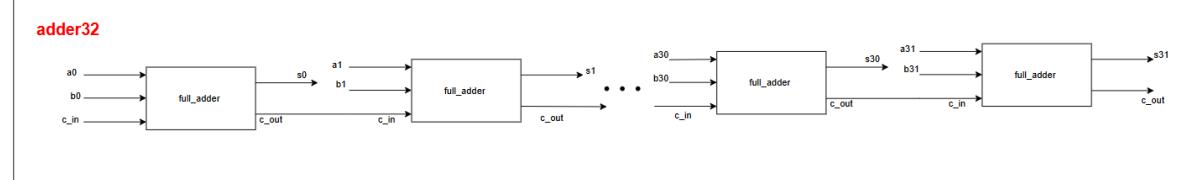
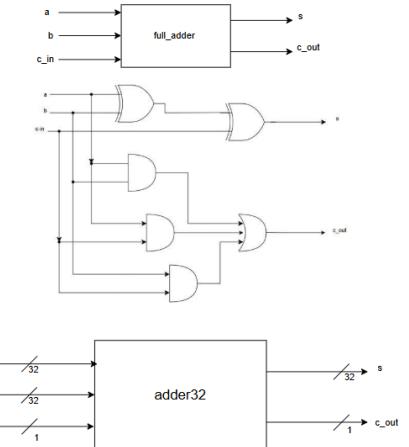
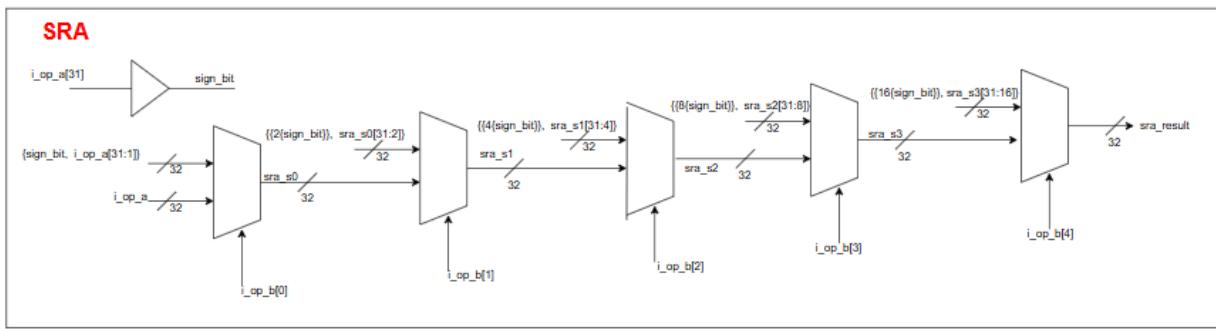
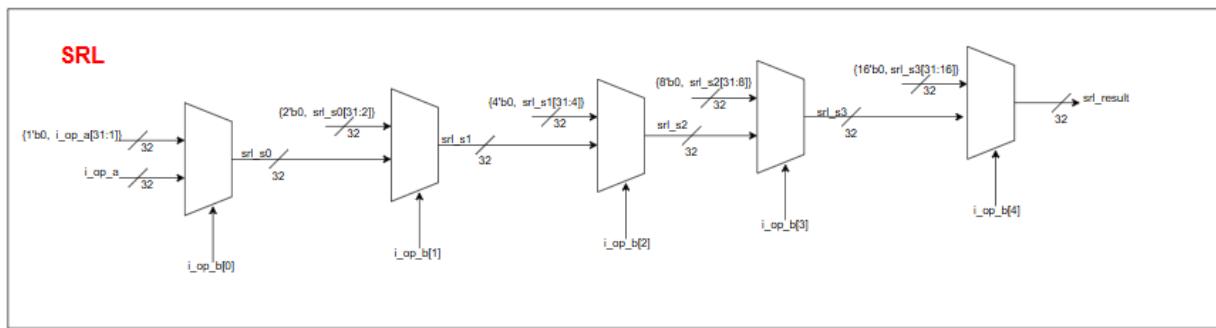
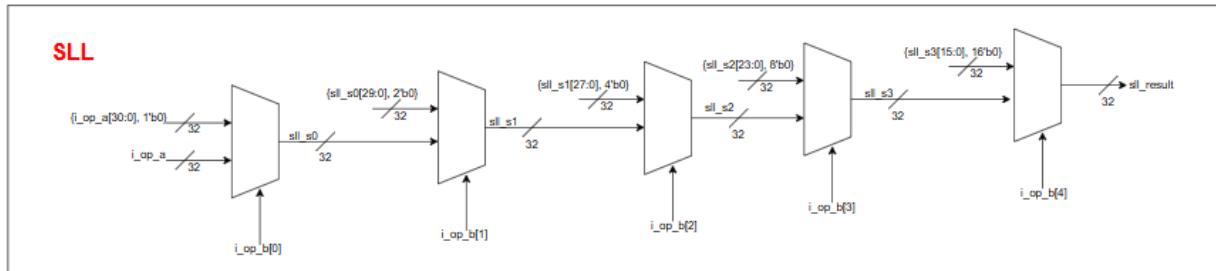
- Thực hiện các phép toán số học và logic dựa trên tín hiệu điều khiển alu_op từ Control Unit
- Nhận giá trị từ Regfile, ImmGen hoặc PC, tính toán và xuất kết quả
- Đặc điểm:** Không dùng các phép toán không thể tổng hợp mà dùng logic

alu_op	R-Type	I-Type
ADD	$rd \leftarrow rs1 + rs2$	$rd \leftarrow rs1 + imm$
SUB	$rd \leftarrow rs1 - rs2$	n/a
SLT	$rd \leftarrow (rs1 < rs2)? 1: 0$	$rd \leftarrow (rs1 < imm)? 1: 0$
SLTU	$rd \leftarrow (rs1 < rs2)? 1: 0$	$rd \leftarrow (rs1 < imm)? 1: 0$
XOR	$rd \leftarrow rs1 \oplus rs2$	$rd \leftarrow rs1 \oplus imm$
OR	$rd \leftarrow rs1 \vee rs2$	$rd \leftarrow rs1 \vee imm$
AND	$rd \leftarrow rs1 \wedge rs2$	$rd \leftarrow rs1 \wedge imm$
SLL	$rd \leftarrow rs1 \ll rs2[4: 0]$	$rd \leftarrow rs1 \ll immrs2[4: 0]$
SRL	$rd \leftarrow rs1 \gg rs2[4: 0]$	$rd \leftarrow rs1 \gg imm[4: 0]$
SRA	$rd \leftarrow rs1 \ggg rs2[4: 0]$	$rd \leftarrow rs1 \ggg imm[4: 0]$

Specifications

Signal	Width	Direction	Description
i_op_a	32	Input	Toán hạng đầu cho ALU
i_op_b	32	Input	Toán hạng thứ hai cho ALU
i_alu_op	4	Input	Tín hiệu điều khiển chọn phép toán sẽ thực hiện
o_alu_data	32	Output	Kết quả trả về từ ALU

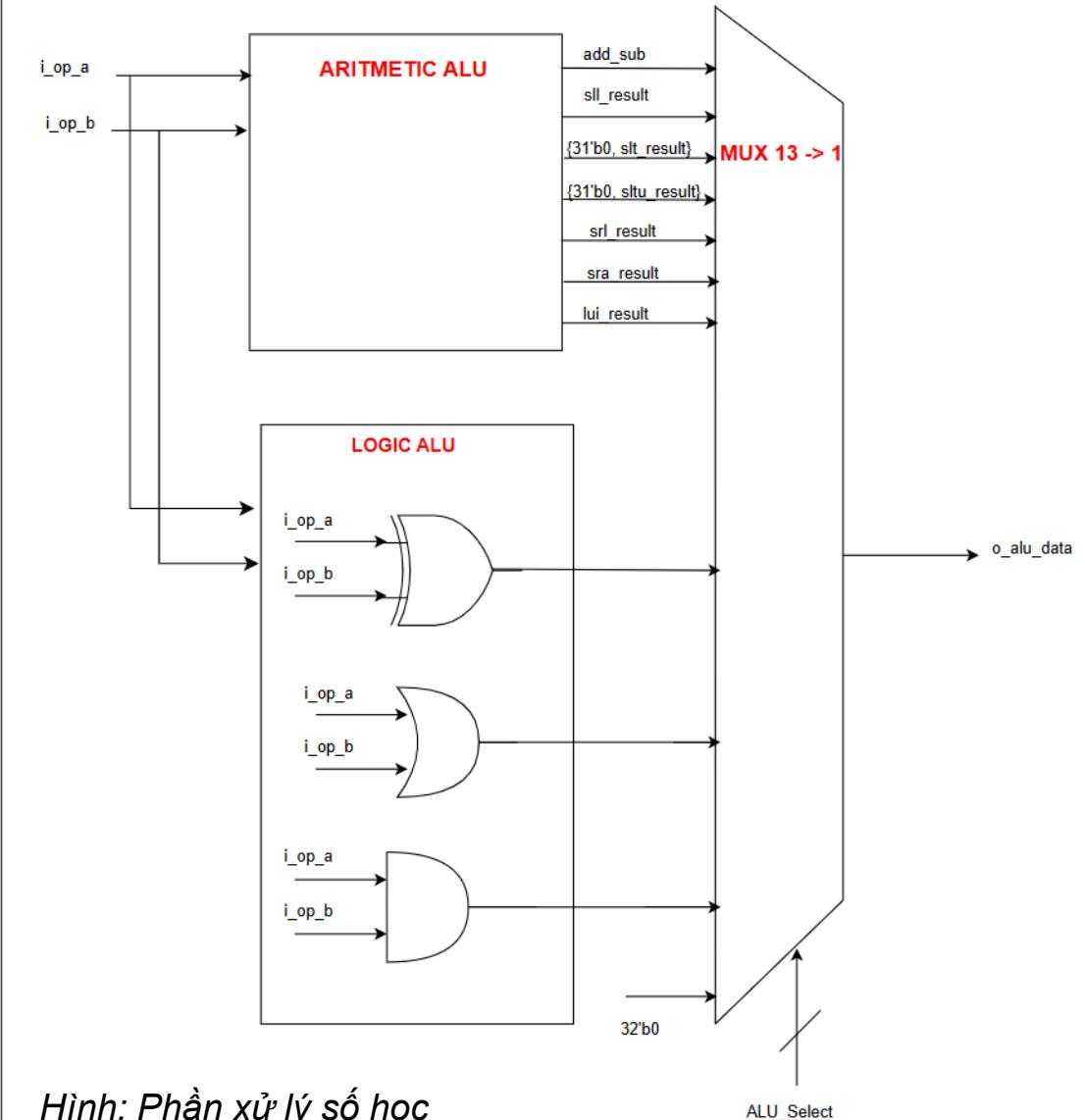
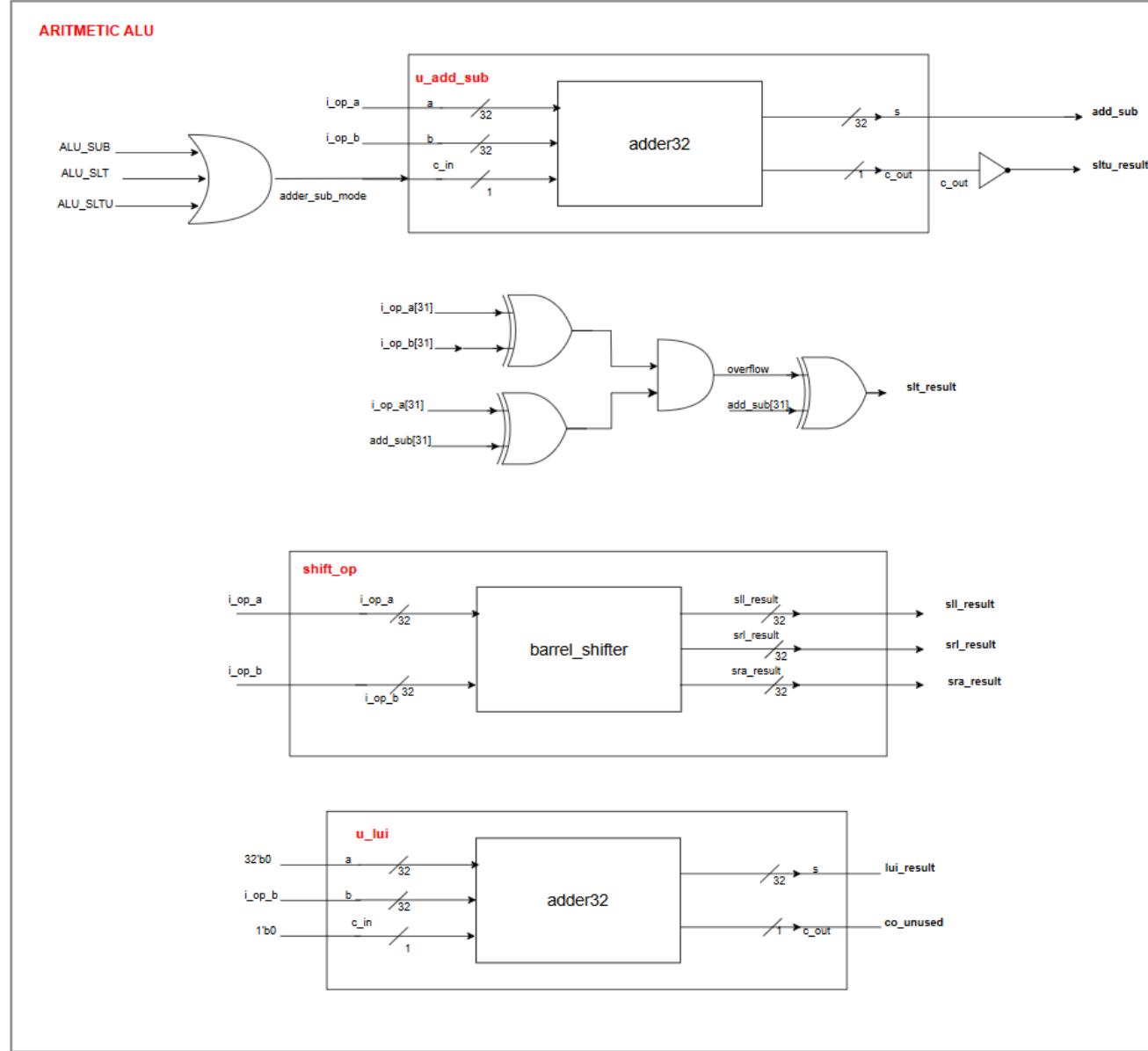
Sơ đồ khối các module hỗ trợ



Hình: Khối Adder-32bit

Hình:
Khối Barrel Shifter

Sơ đồ khối của ALU



Hình: Phần xử lý số học và sơ đồ khối của ALU

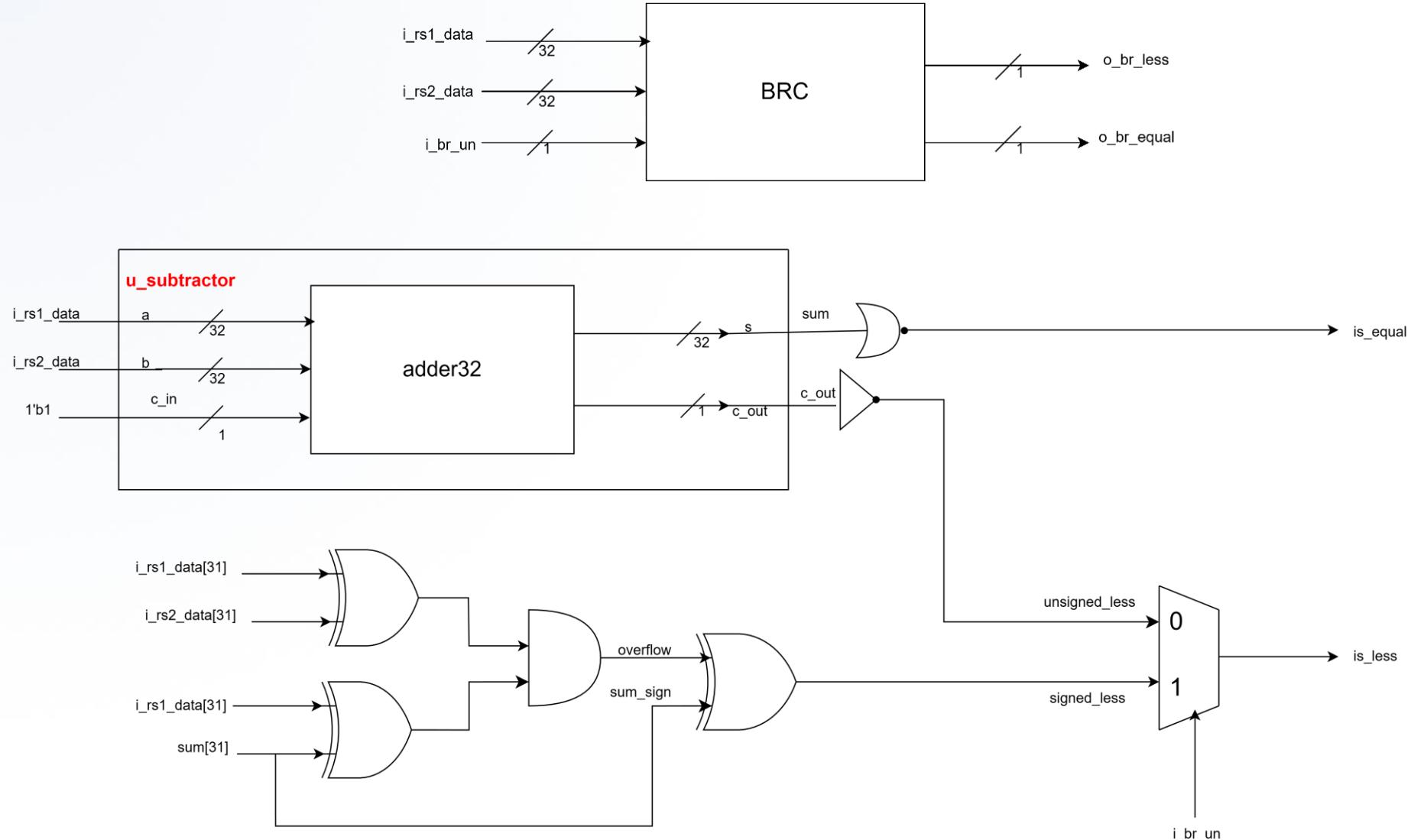
Khối BRC (Branch Comparison Unit)

- ▶ Thực hiện so sánh hai toán hạng
- ▶ Xử lý dữ liệu có dấu (Signed) và không dấu (Unsigned)
- ▶ Ngõ ra là ngõ vào lựa chọn của PC (pc_sel)

Specifications

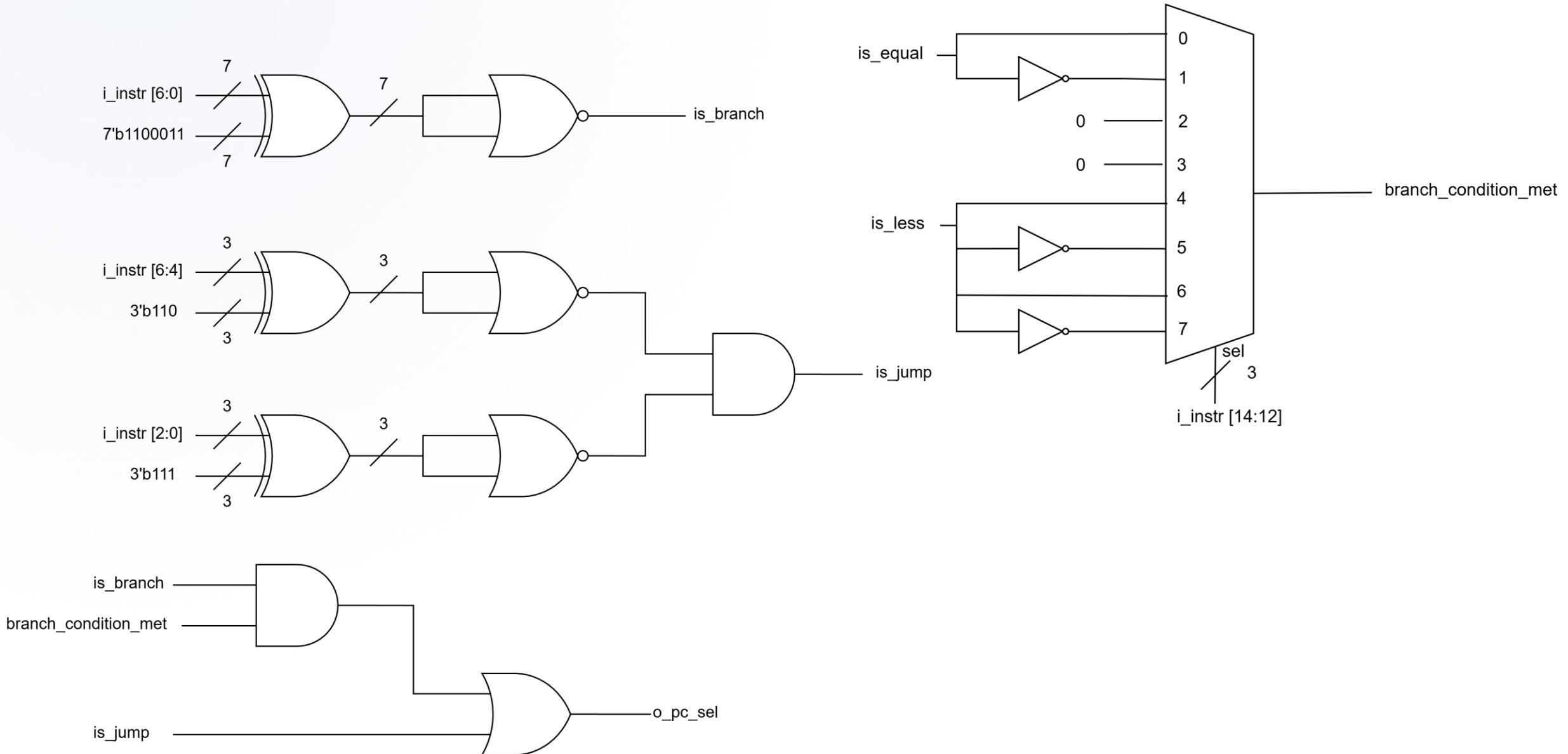
Signal name	Width	Direction	Description
i_rs1_data	32	input	Data from the first source register
i_rs2_data	32	input	Data from the second source register
i_br_un	1	input	Comparison mode (1 if signed, 0 if unsigned).
i_instr	3	input	Instruction from EX Stage
o_pc_sel	1	input	PC selection Signal

Sơ đồ khối của BRC



Hình 8: Sơ đồ chi tiết bên trong BRC

Sơ đồ khối của BRC



Khối ImmGen (Immediate Generator)

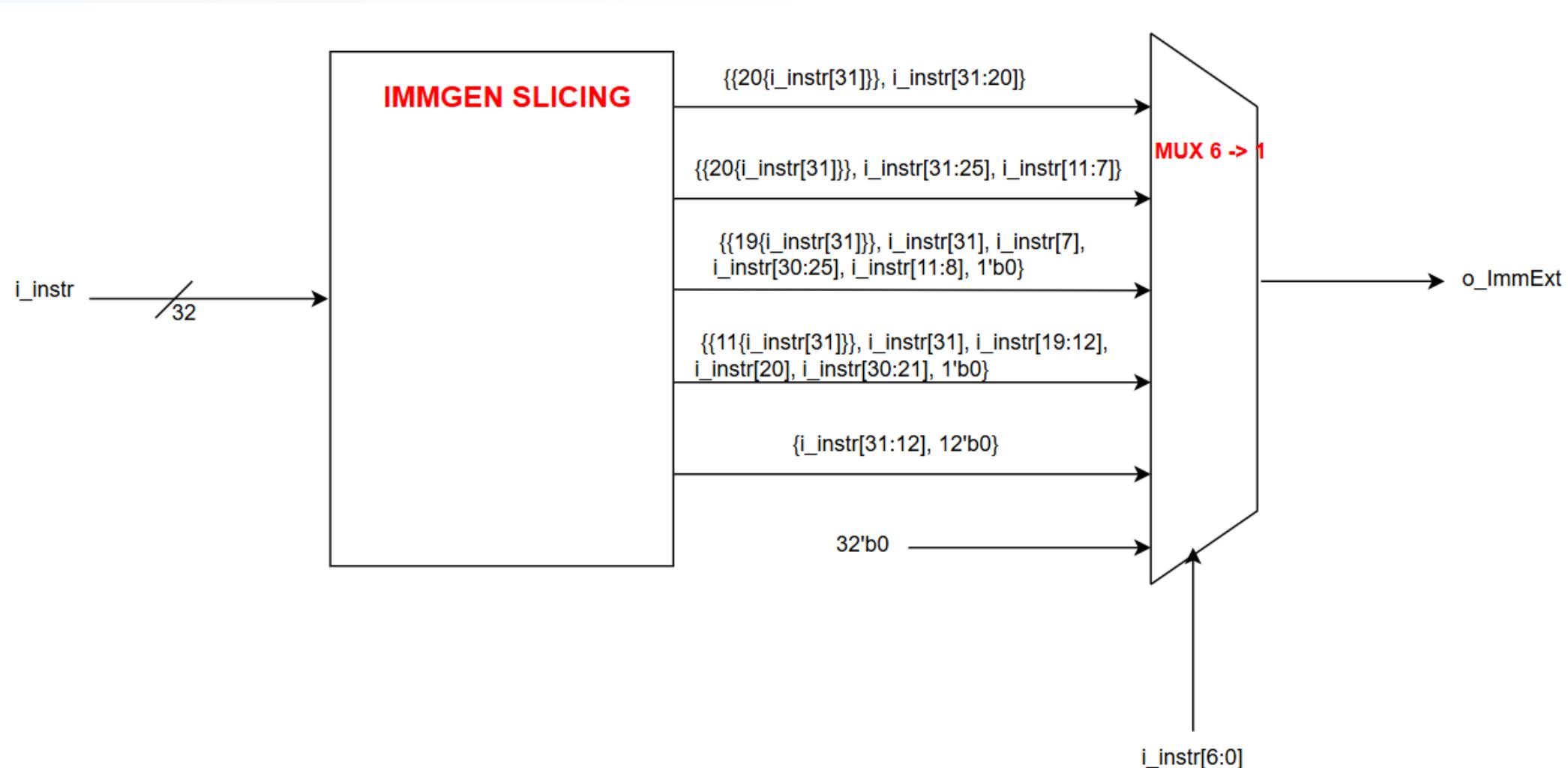
- ▶ Tách và mở rộng dấu (sign-extend) các giá trị immediate từ Instruction
- ▶ Khối thuần tổ hợp tái sắp xếp các bit theo 5 định dạng: I, S, B, U, J-Type.

Cách xử lý từng định dạng

Type	Immediate
I-Type	$\{\{20\{\text{instr}[31]\}\}, \text{instr}[31:20]\}$
S-Type	$\{\{20\{\text{instr}[31]\}\}, \text{instr}[31:25], \text{instr}[11:7]\}$
B-Type	$\{\{19\{\text{instr}[31]\}\}, \text{instr}[31], \text{instr}[7], \text{instr}[30:25], \text{instr}[11:8], 1'b0\}$
J-Type	$\{\{11\{\text{instr}[31]\}\}, \text{instr}[31], \text{instr}[19:12], \text{instr}[20], \text{instr}[30:21], 1'b0\}$
U-Type	$\{\text{instr}[31:12], 12'b0\}$

Specifications	Signal	Width	Direction	Description
	i_instr	32	input	Mã lệnh 32 bit cho biết cấu trúc câu lệnh
	o_immExt	32	output	Giá trị immediate ngõ ra tương ứng từng định dạng

Sơ đồ khối của ImmGen



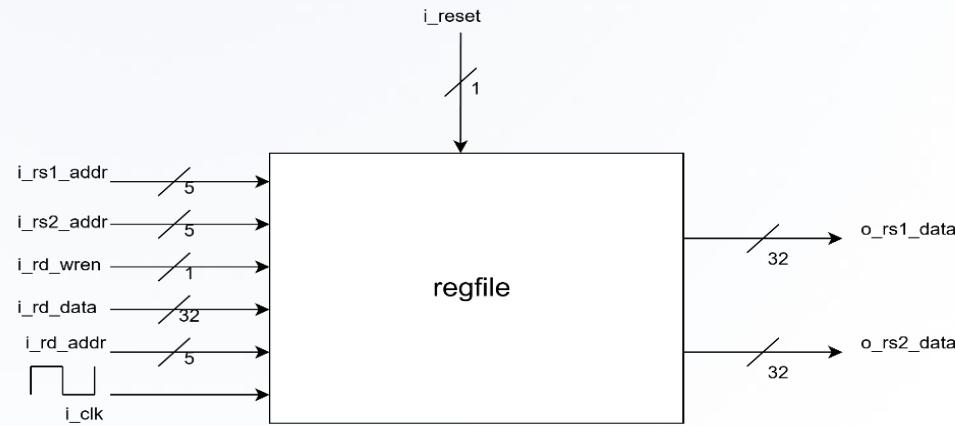
Hình 9: Tổng quan khối ImmGen

Khối RegFile (Register File)

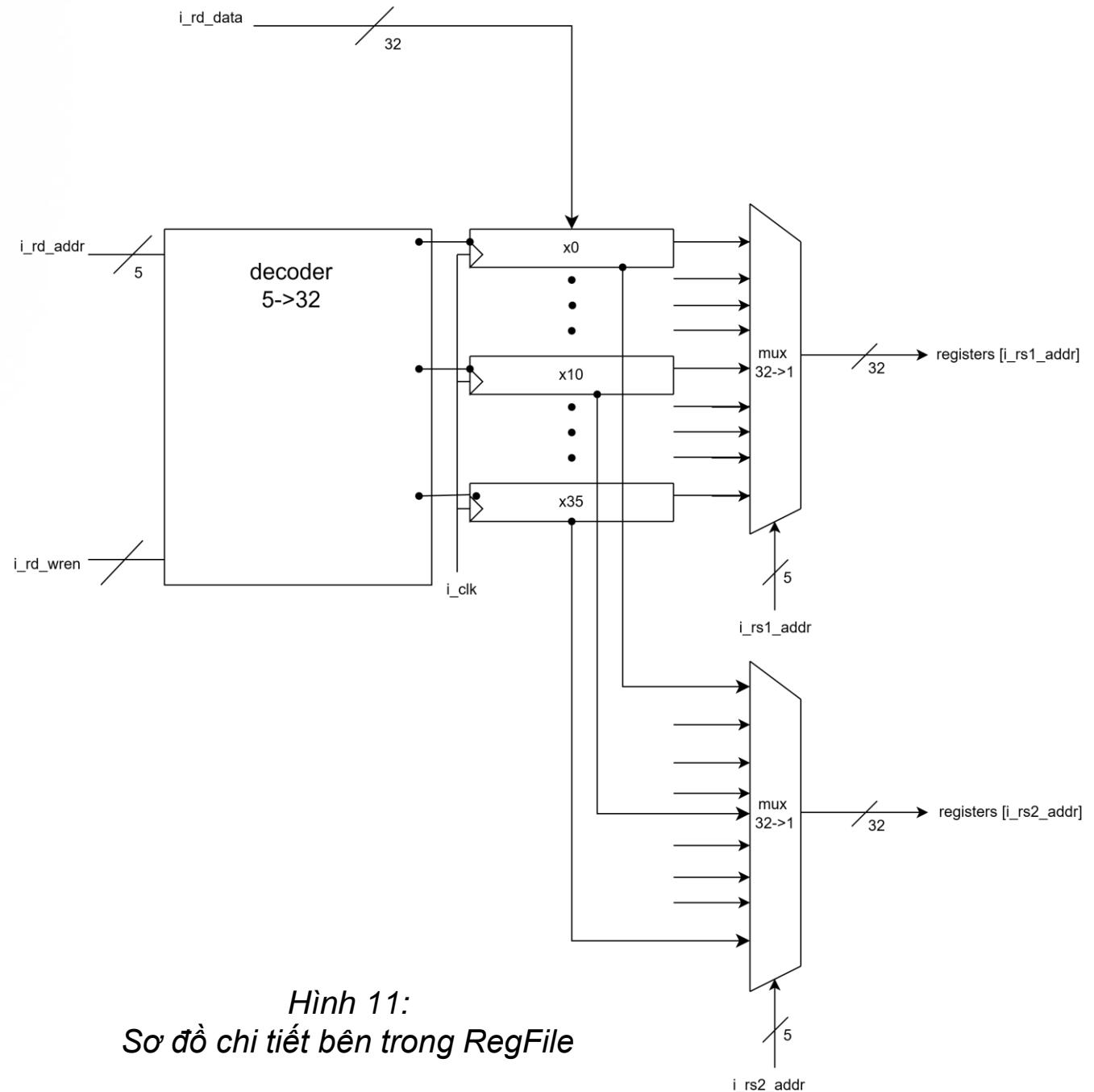
- Lưu trữ tạm thời các thanh ghi và kết quả
- Hỗ trợ hai cổng đọc không đồng bộ và một cổng ghi đồng bộ
- Có hỗ trợ đọc sau khi ghi (Read After Write – RAW)

Specifications	Signal name	Width	Direction	Description
	i_clk	1	input	global clock
	i_reset	1	input	global reset
	i_rs1_addr	5	input	address of the first source register.
	i_rs2_addr	5	input	address of the second source register.
	o_rs1_data	32	output	Data from the first source register
	o_rs2_data	32	output	Data from the second source register
	i_rd_addr	5	input	Address of the destination register
	i_rd_data	32	output	Data to write to the destination register
	i_rd_wren	1	input	Write enable for the destination register

Sơ đồ khối của RegFile

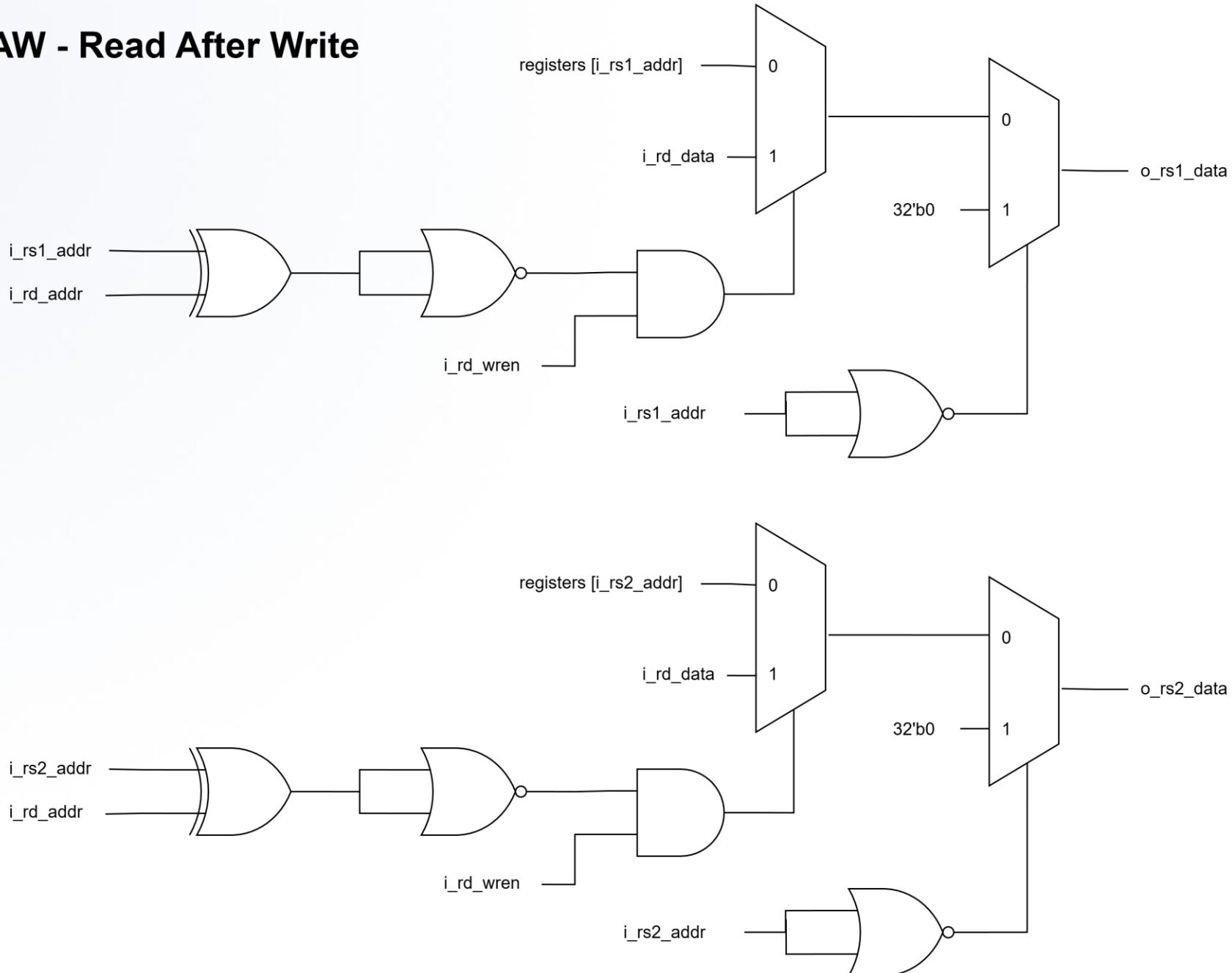


Hình 10: Tổng quan các ngõ vào/ra của khối RegFile



Hình 11:
Sơ đồ chi tiết bên trong RegFile

RAW - Read After Write



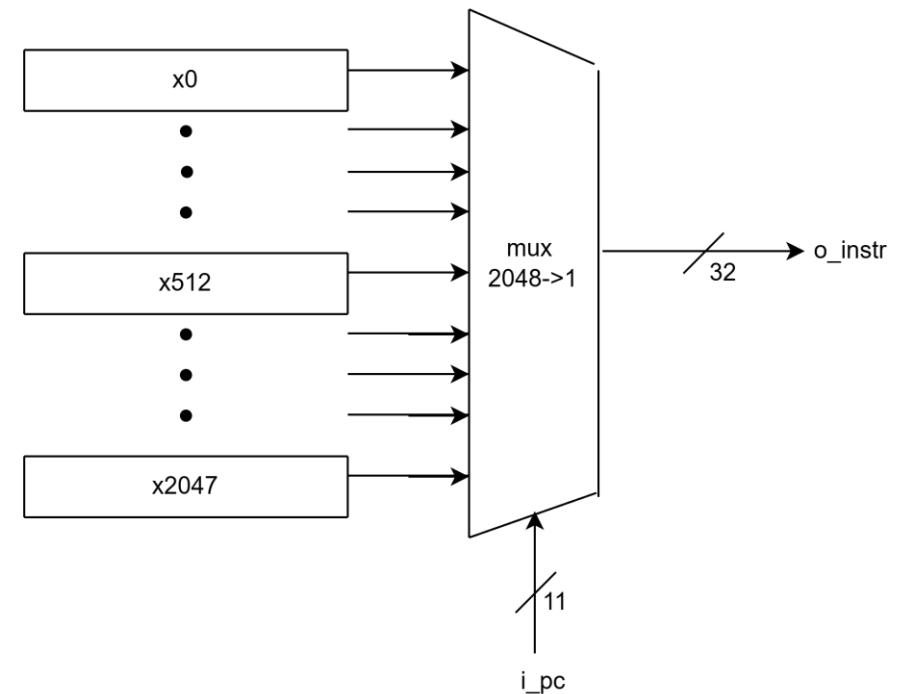
Khối IMEM (Instruction Memory)

- ▶ Lưu trữ chương trình lệnh, chỉ thực hiện chức năng đọc
- ▶ Trả mã lệnh tương ứng với địa chỉ nhận được

Specifications

Signal	Width	Direction	Description
i_pc	32	input	Địa chỉ lệnh được lấy ra
o_instr	32	output	Lệnh 32-bit sẽ thực thi

Sơ đồ khối



Khối LSU (Load-Store Unit)

- ▶ Truy cập bộ nhớ dữ liệu (Data Memory) và giao tiếp ngoại vi (Peripheral devices)
- ▶ Xử lý các lệnh Load và Store

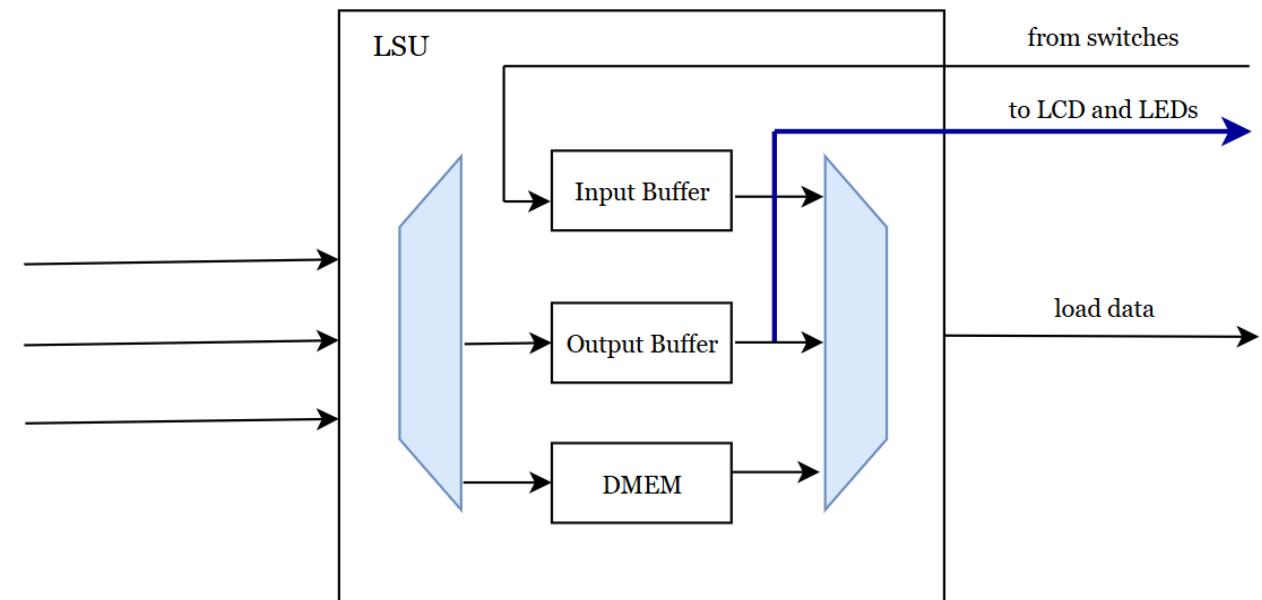
Specifications	Signal	Width	Direction	Description
	i_clk	1	input	Xung clock kích cạnh lên
	i_reset	1	input	Tín hiệu reset
	i_lsu_addr	32	input	Địa chỉ đọc ghi data
	i_funct3	3	input	Chọn kích thước đọc ghi
	i_st_data	32	input	Dữ liệu được ghi vào LSU
	i_lsu_wren	1	input	Tín hiệu cho phép ghi (1 = được phép ghi)
	o_ld_data	32	output	Dữ liệu đọc từ LSU
	o_io_ledr	32	output	Dữ liệu xuất ra Red LEDs
	o_io_leg	32	output	Dữ liệu xuất ra Green LEDs
	o_io_hex0..7	7	output	Dữ liệu xuất ra 7-segments LEDs
	o_io_lcd	32	output	Dữ liệu xuất ra LCDs
	i_io_sw	32	input	Dữ liệu nhận từ Switches

Khối LSU (Load-Store Unit)

Phân vùng địa chỉ (Memory Mapping) và Sơ đồ khối LSU

Base address	Top address	Mapping
0x1001_1000	0xFFFF_FFFF	(Reserved)
0x1001_0000	0x1001_0FFF	Switches
0x1000_5000	0x1000_FFFF	(Reserved)
0x1000_4000	0x1000_4FFF	LCD Control Registers
0x1000_3000	0x1000_3FFF	7-segment LEDs 7-4
0x1000_2000	0x1000_2FFF	7-segment LEDs 3-0
0x1000_1000	0x1000_1FFF	Green LEDs
0x1000_0000	0x1000_0FFF	Red LEDs
0x0001_0000	0x0FFF_FFFF	(Reserved)
0x0000_0000	0x0000_FFFF	Memory (64KiB)

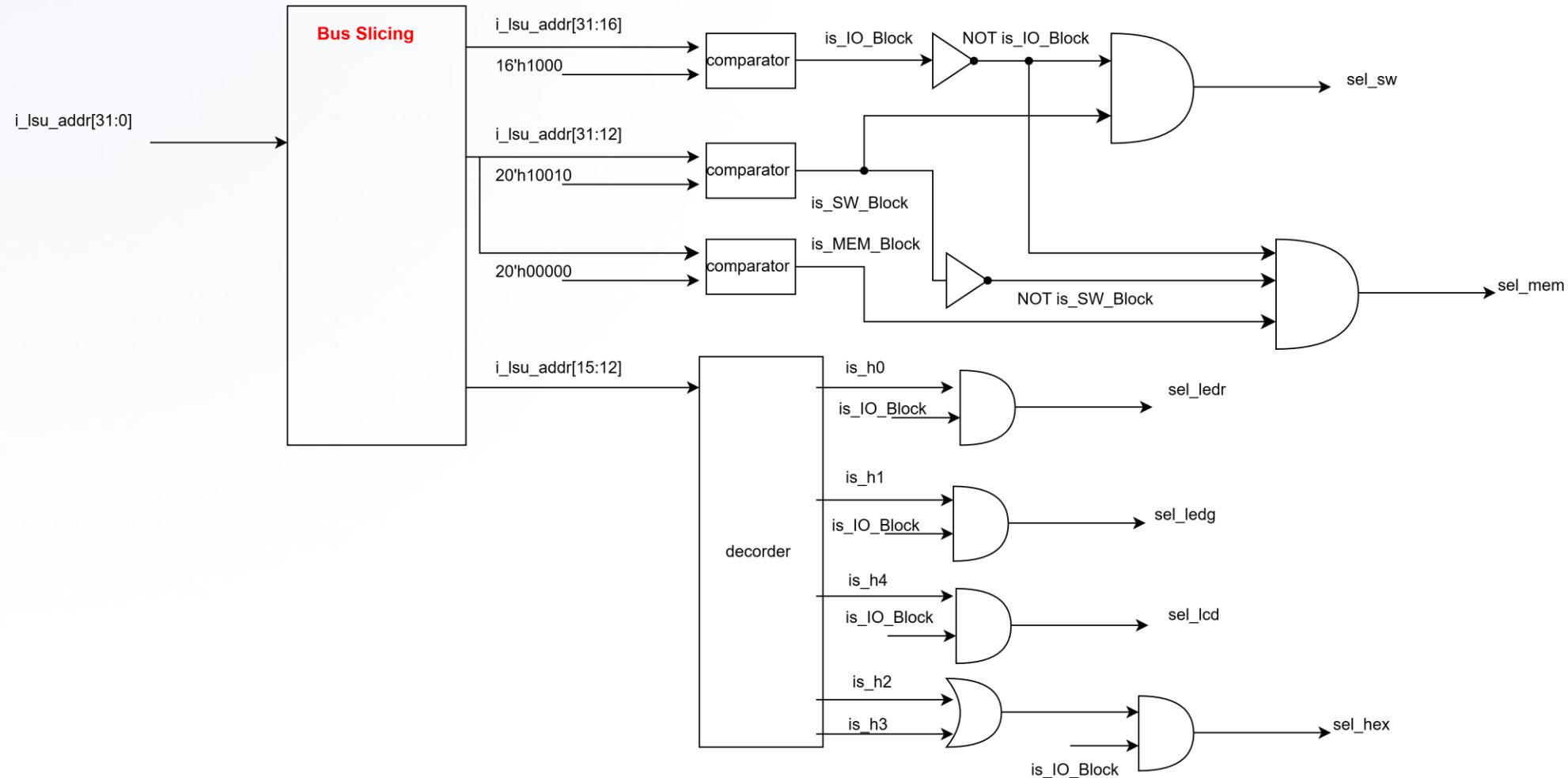
Bảng: Bảng phân vùng địa chỉ (Memory Mapping) các ngoại vi



Hình: Tổng quát LSU

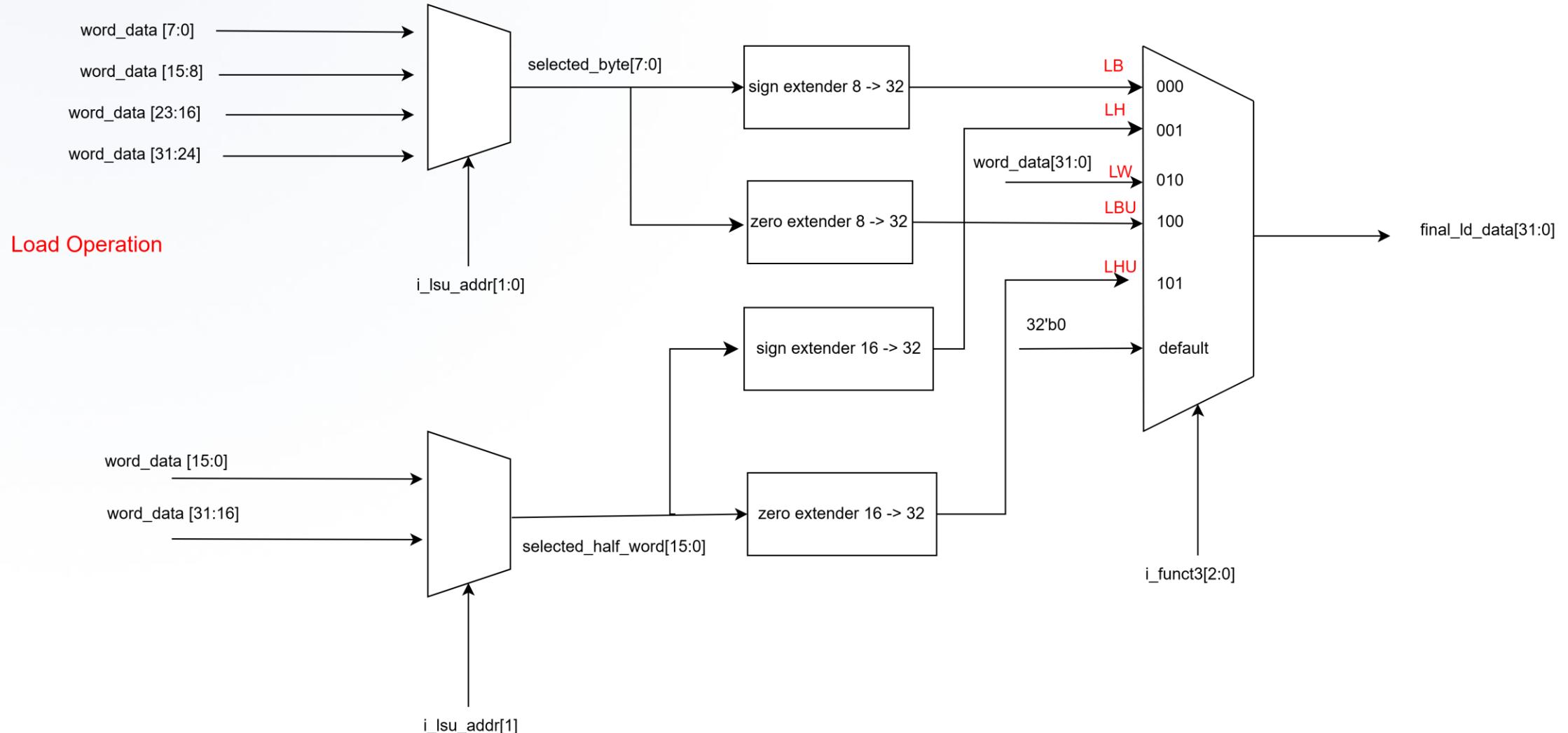
Sơ đồ khối của LSU

Khối giải mã địa chỉ (Address Decoder)



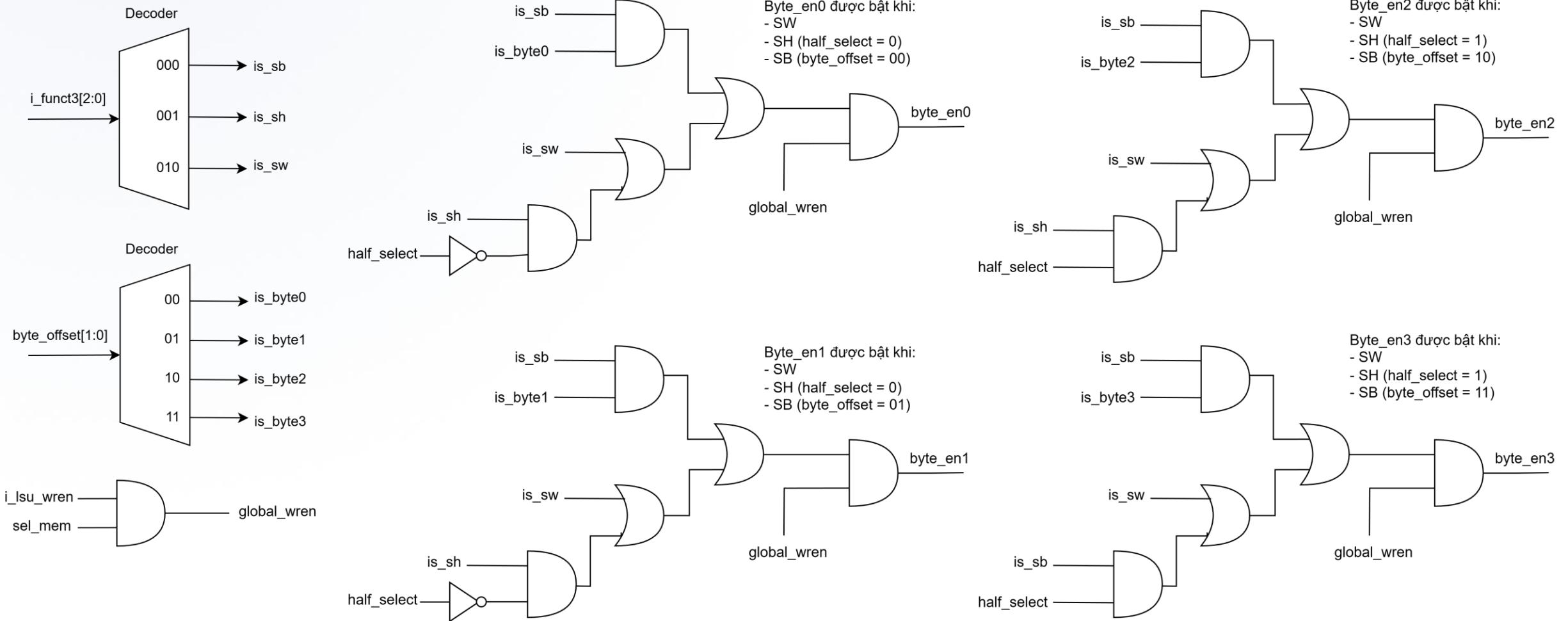
Sơ đồ khối của LSU

Khối đọc bộ nhớ (Memory Read) – Nhóm lệnh Load – Hỗ trợ đọc đồng bộ



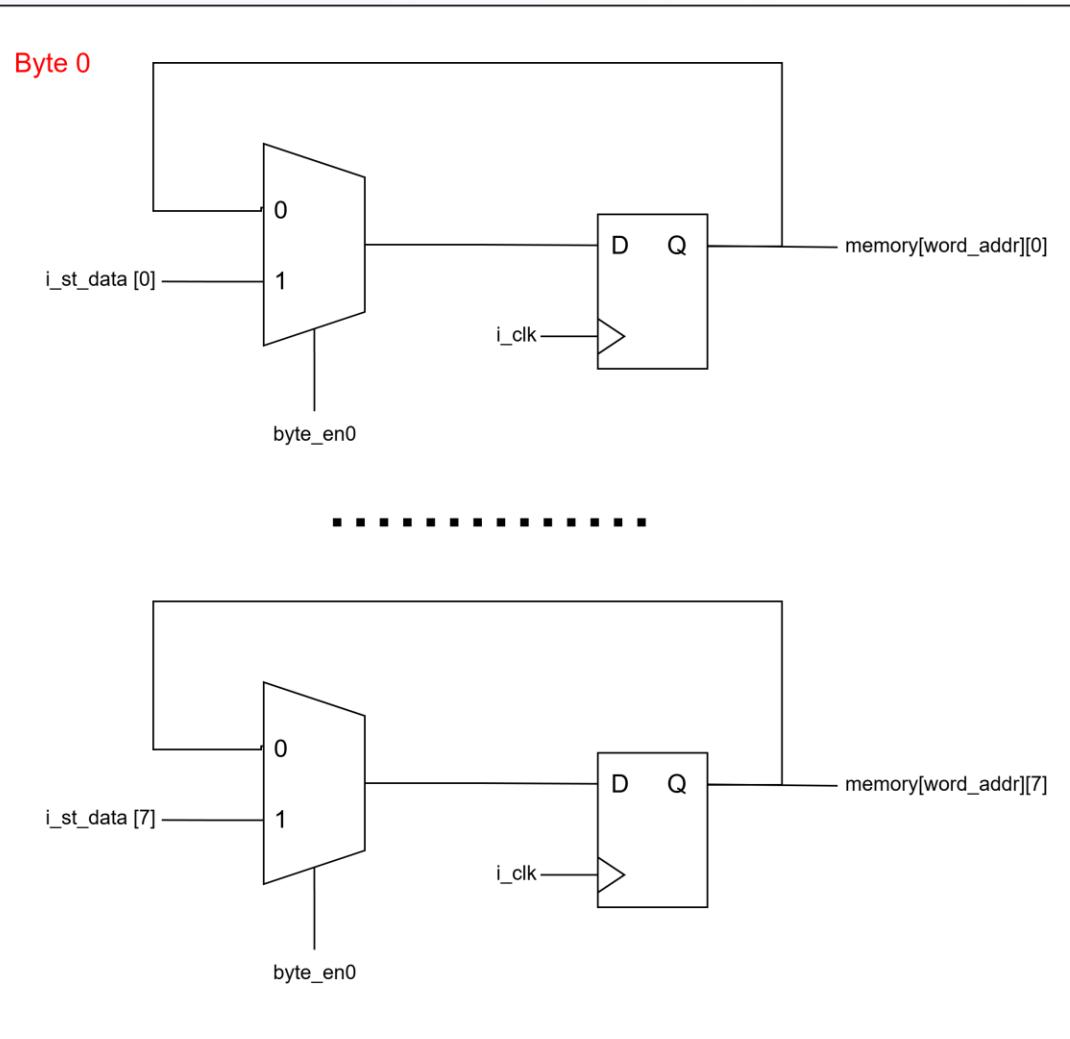
Sơ đồ khối của LSU

Khối đọc bộ nhớ (Memory Read) – Nhóm lệnh Store



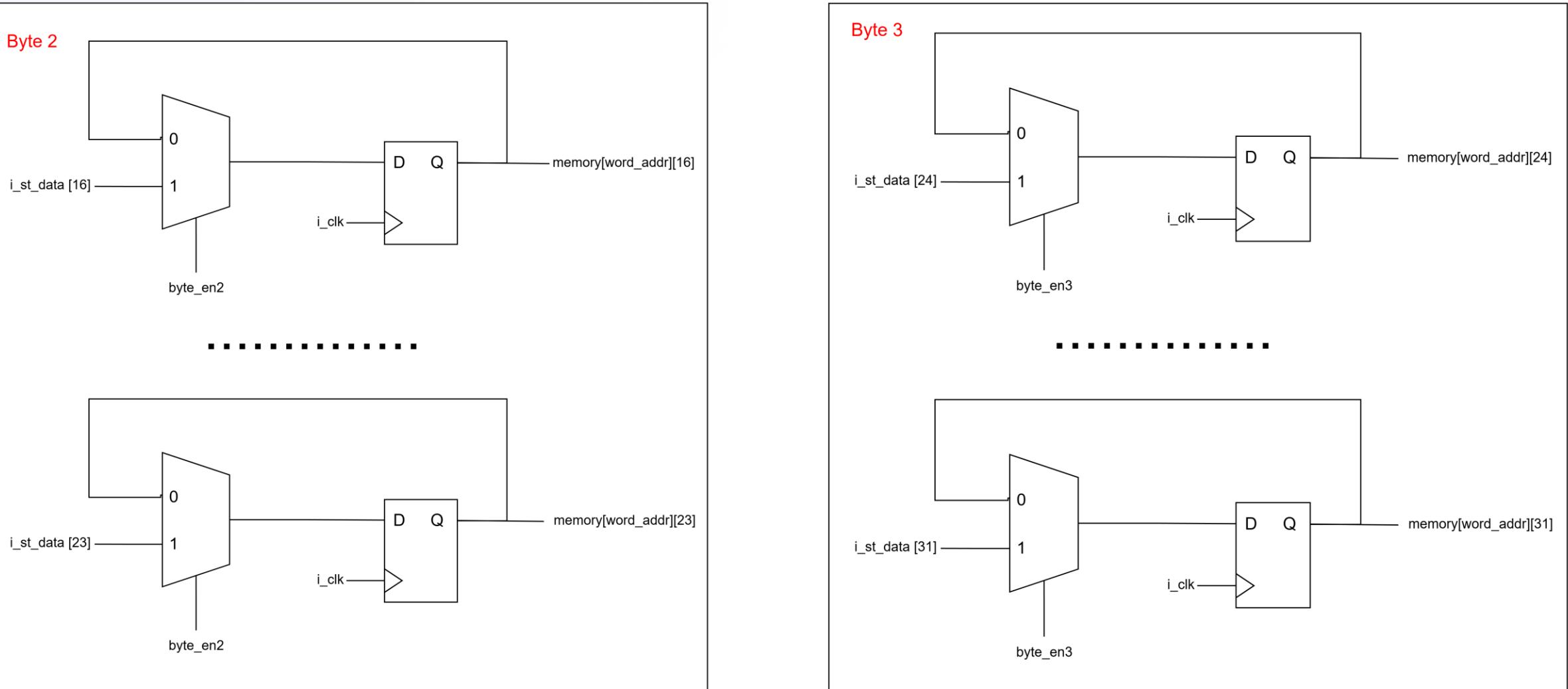
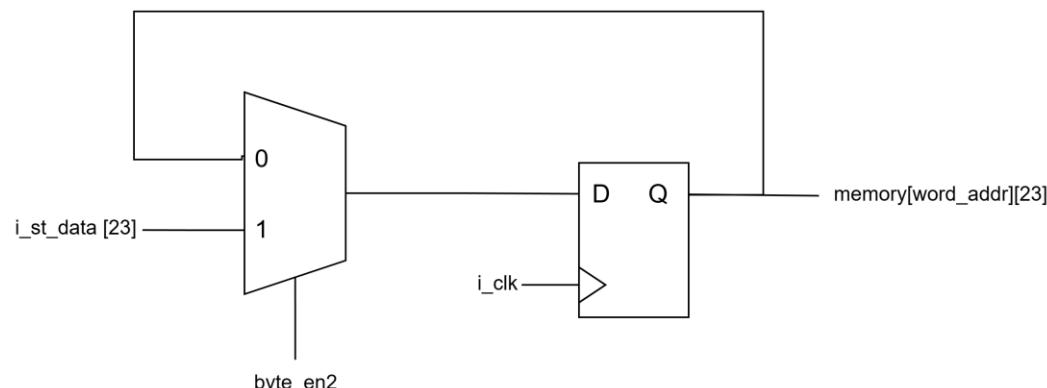
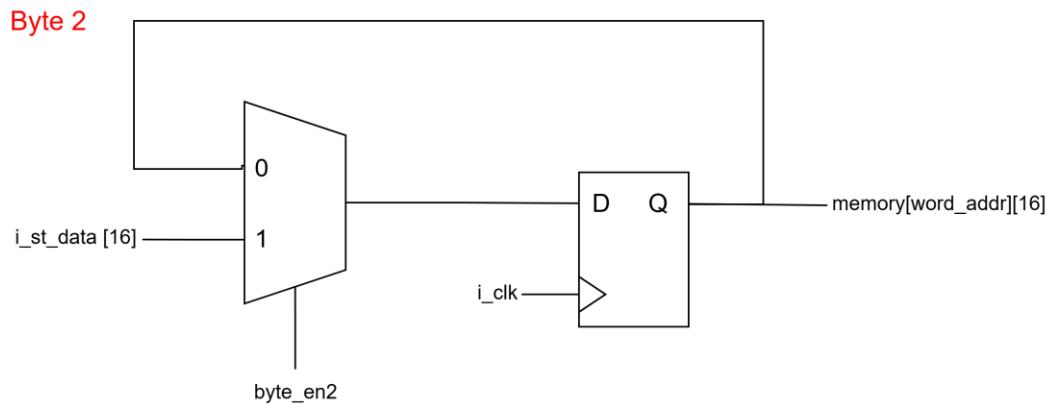
Sơ đồ khối của LSU

Khối đọc bộ nhớ (Memory Read) – Nhóm lệnh Store



Sơ đồ khối của LSU

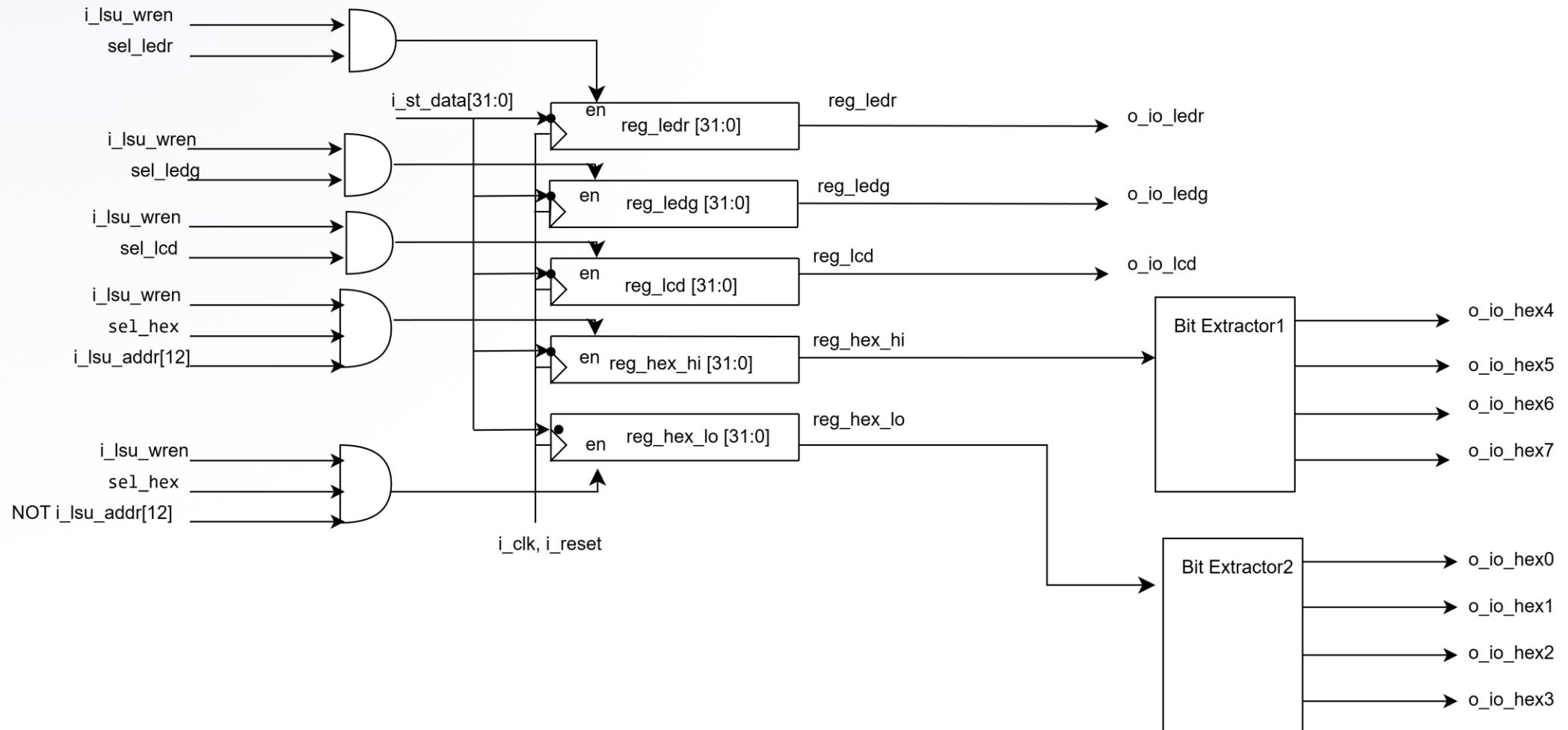
Khối đọc bộ nhớ (Memory Read) – Nhóm lệnh Store



Sơ đồ khối của LSU

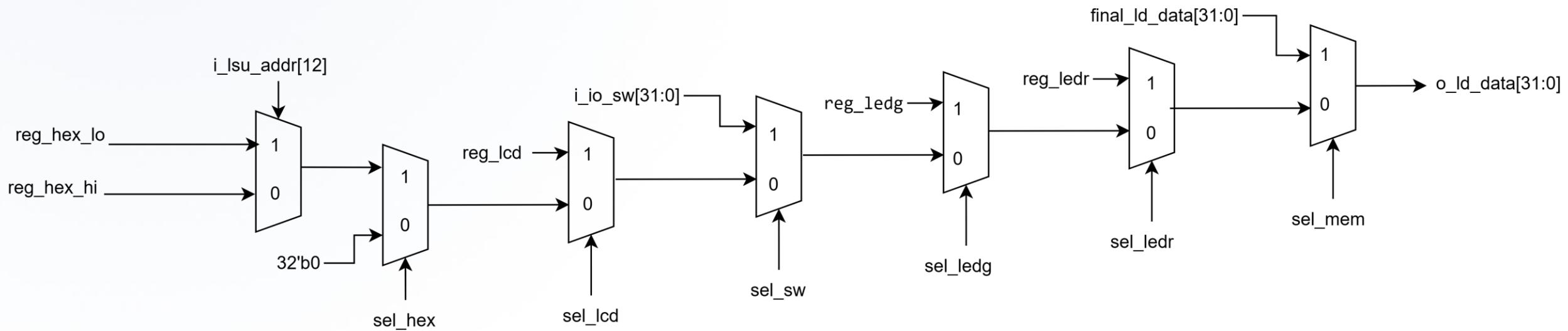
Khối các thanh ghi I/O – I/O registers

Khối Thanh ghi I/O



Sơ đồ khối của LSU

Khối đọc dữ liệu ghép kênh (Read data multiplexing)



Khối Control Unit

- Thiết kế theo phương pháp Hardwired Control
- Nằm ở pha ID, nhận lệnh vào và xuất các tín hiệu điều khiển

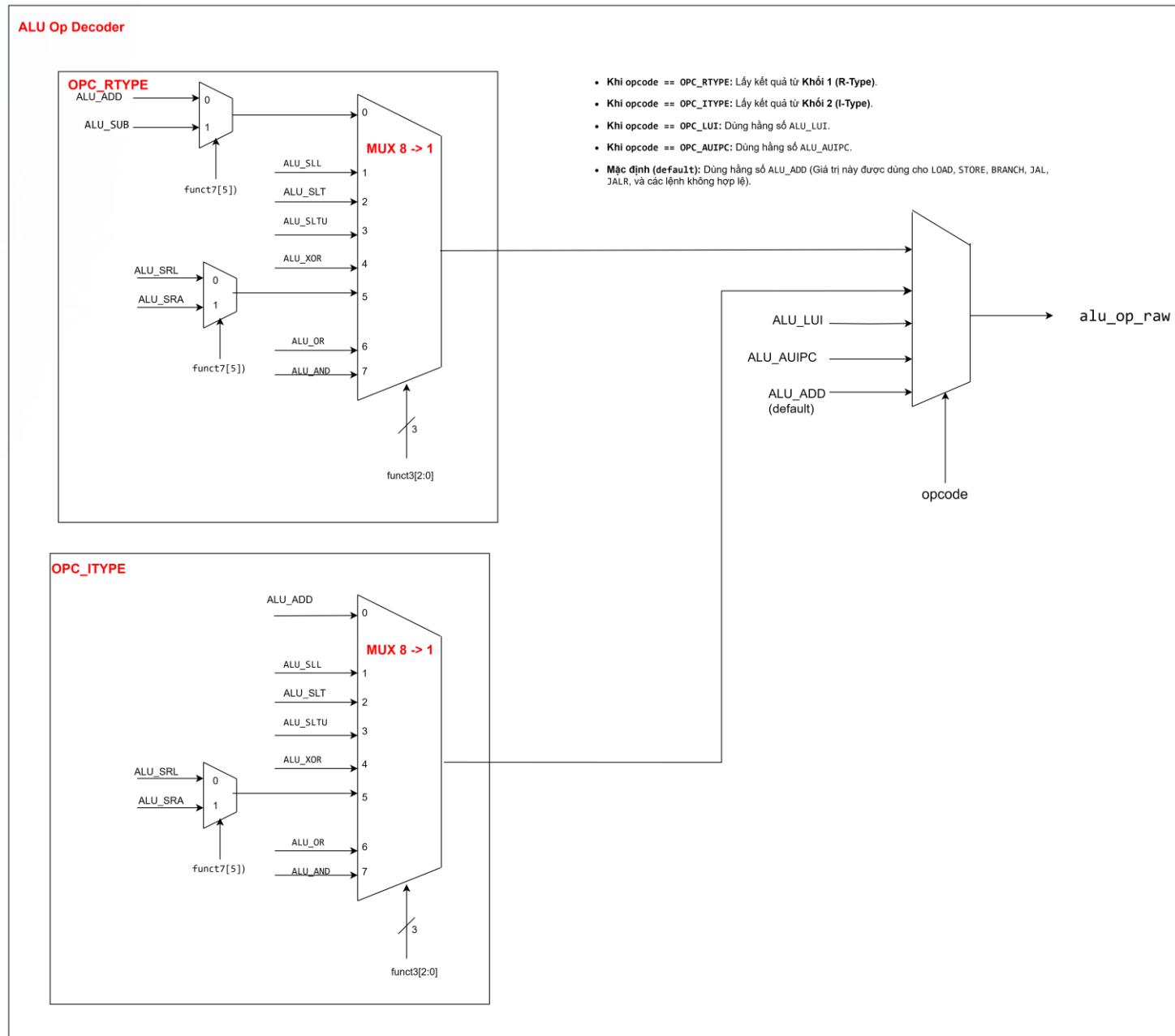
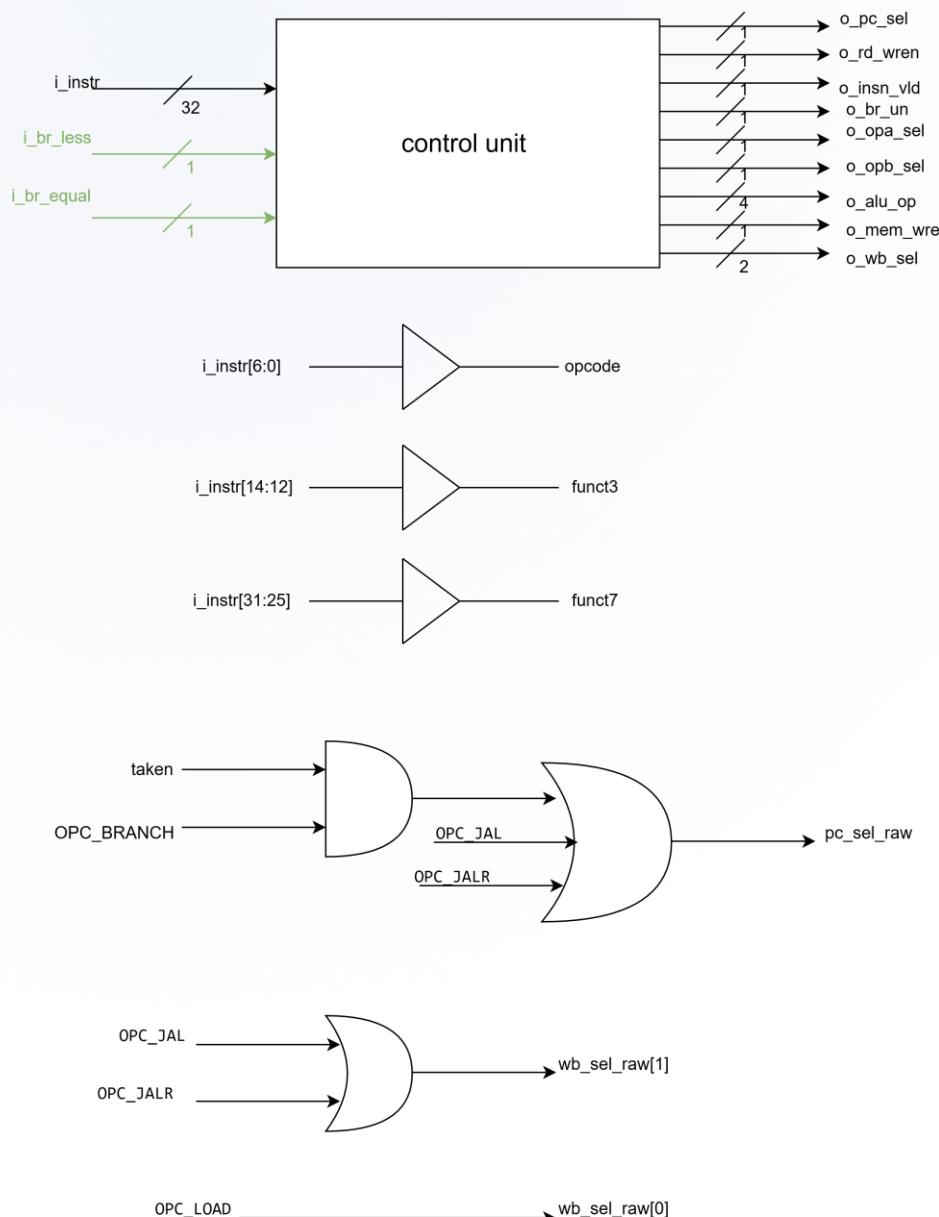
Specifications	Signal	Width	Direction	Description
	i_instr	32	input	Mã lệnh lấy từ IMEM (sử dụng opcode, function 3, function 7)
	i_br_less	1	input	Kết quả so sánh bé từ BRC
	i_br_equal	1	input	Kết quả so sánh bằng từ BRC
	o_pc_sel	1	output	Tín hiệu chọn PC
	o_rd_wren	1	output	Tín hiệu cho phép ghi vào thanh ghi đích
	o_insn_vld	1	output	Tín hiệu báo lệnh hợp lệ
	o_br_un	1	output	Tín hiệu chọn mode so sánh (1=signed; 0=unsigned)
	o_opa_sel	1	output	Tín hiệu chọn toán hạng a cho ALU
	o_opb_sel	1	output	Tín hiệu chọn toán hạng b cho ALU
	o_alu_op	4	output	Tín hiệu chọn phép toán cho ALU
	o_mem_wren	1	output	Tín hiệu cho phép ghi vào LSU
	o_wb_sel	2	output	Tín hiệu chọn dữ liệu ghi vào rd

Bảng chân trị tóm tắt logic giải mã

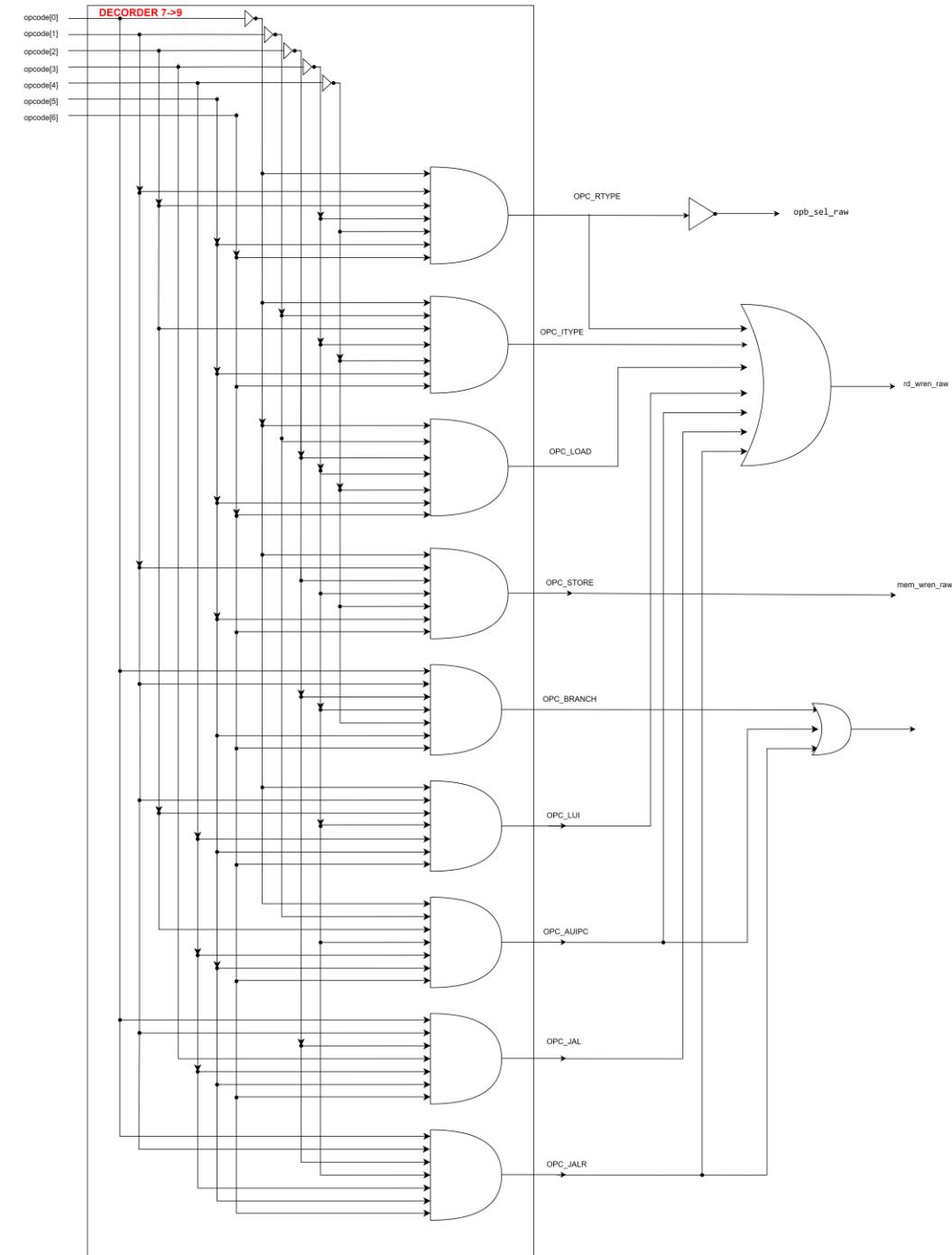
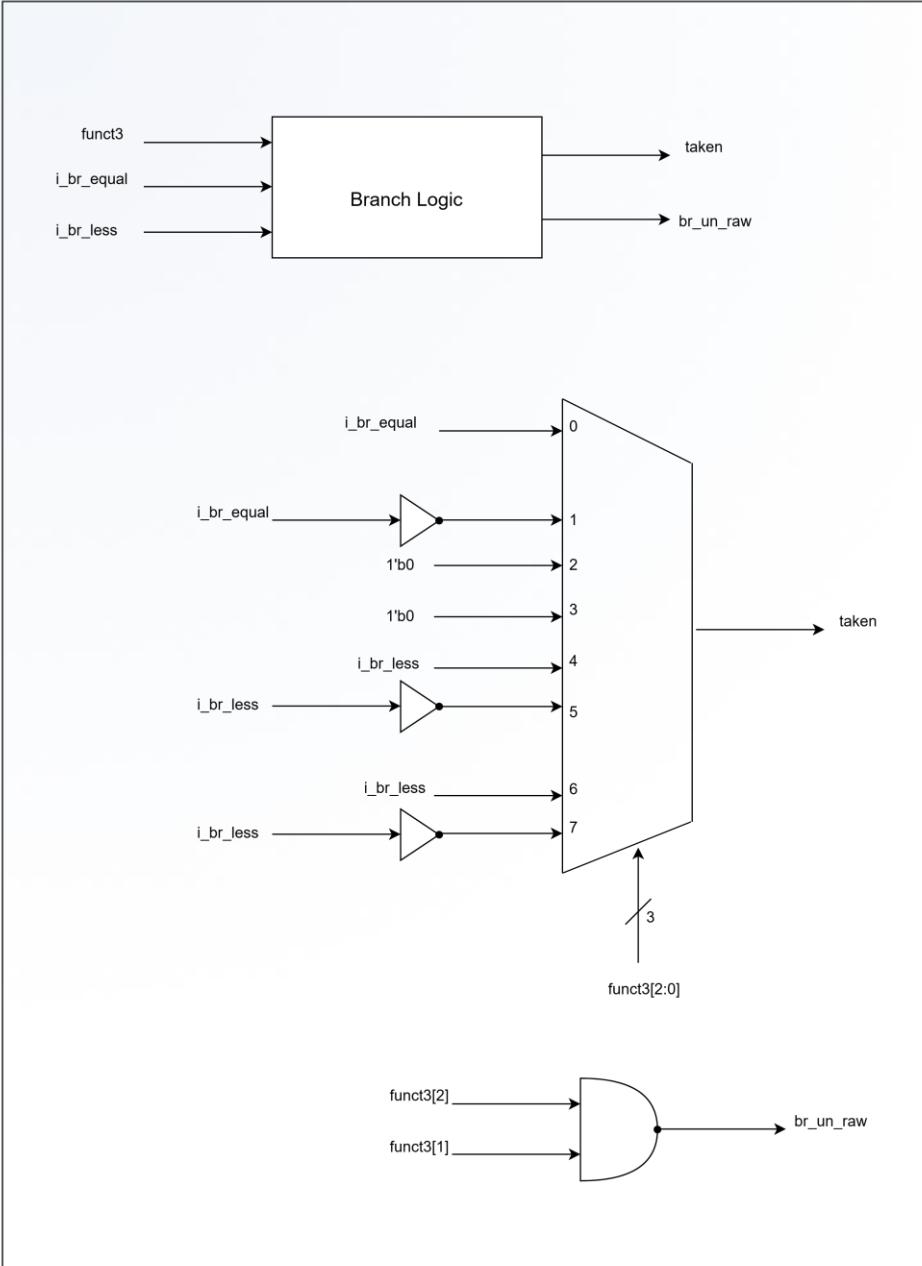
Type		instr[31:25]	instr[14:12]	instr[6:0]	br_less	br_equal	pc_sel	rd_wren	br_un	opa_sel	opb_sel	alu_op	mem_wren	wb_sel	insn_vld
R	ADD	0000000	000	0110011	x	x	0	1	x	0	0	0000	0	00	x
	SUB	0100000	000	0110011	x	x	0	1	x	0	0	0001	0	00	x
	SLL	0000000	001	0110011	x	x	0	1	x	0	0	0010	0	00	x
	SLT	0000000	010	0110011	x	x	0	1	x	0	0	0011	0	00	x
	SLTU	0000000	011	0110011	x	x	0	1	x	0	0	0100	0	00	x
	XOR	0000000	100	0110011	x	x	0	1	x	0	0	0101	0	00	x
	SRL	0000000	101	0110011	x	x	0	1	x	0	0	0110	0	00	x
	SRA	0100000	101	0110011	x	x	0	1	x	0	0	0111	0	00	x
	OR	0000000	110	0110011	x	x	0	1	x	0	0	1000	0	00	x
	AND	0000000	111	0110011	x	x	0	1	x	0	0	1001	0	00	x
I	ADDI	x	000	0010011	x	x	0	1	x	0	1	0000	0	00	x
	SLTI	x	010	0010011	x	x	0	1	x	0	1	0011	0	00	x
	SLTIU	x	011	0010011	x	x	0	1	x	0	1	0100	0	00	x
	XORI	x	100	0010011	x	x	0	1	x	0	1	0101	0	00	x
	ORI	x	110	0010011	x	x	0	1	x	0	1	1000	0	00	x
	ANDI	x	111	0010011	x	x	0	1	x	0	1	1001	0	00	x
	SLLI	0000000	001	0010011	x	x	0	1	x	0	1	0010	0	00	x
	SRLI	0000000	101	0010011	x	x	0	1	x	0	1	0110	0	00	x
	SRAI	0100000	101	0010011	x	x	0	1	x	0	1	0111	0	00	x
LOAD	LB	x	000	0000011	x	x	0	1	x	0	1	0000	0	01	x
	LH	x	001	0000011	x	x	0	1	x	0	1	0000	0	01	x
	LW	x	010	0000011	x	x	0	1	x	0	1	0000	0	01	x
	LBU	x	100	0000011	x	x	0	1	x	0	1	0000	0	01	x
	LHW	x	101	0000011	x	x	0	1	x	0	1	0000	0	01	x
STORE	SB	x	000	0100011	x	x	0	0	x	0	1	0000	1	x	x
	SH	x	001	0100011	x	x	0	0	x	0	1	0000	1	x	x
	SW	x	010	0100011	x	x	0	0	x	0	1	0000	1	x	x
BRANCH	BEQ	x	000	1100011	x	1	1	0	x	1	1	0000	0	x	x
						0	0								
	BNE	x	001	1100011	x	0	1	0	x	1	1	0000	0	x	x
						1	0								
	BLT	x	100	1100011	1	x	1	0	1	1	1	0000	0	x	x
						0	0								
	BGE	x	101	1100011	0	x	1	0	1	1	1	0000	0	x	x
U	BLTU	x	110	1100011	1	x	1	0	0	1	1	0000	0	x	x
						0	0								
	BGEU	x	111	1100011	0	x	1	0	0	1	1	0000	0	x	x
U	LUI	x	x	0110111	x	x	0	1	x	x	1	1010	0	00	x
	AUIPC	x	x	0010111	x	x	0	1	x	1	1	1011	0	00	x
J	JAL	x	x	1101111	x	x	1	1	x	1	1	0000	0	10	x
	JALR	x	000	1100111	x	x	1	1	x	0	1	0000	0	10	x

Chú thích: Các giá trị x là Không quan trọng

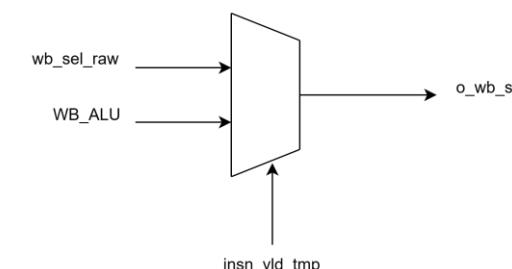
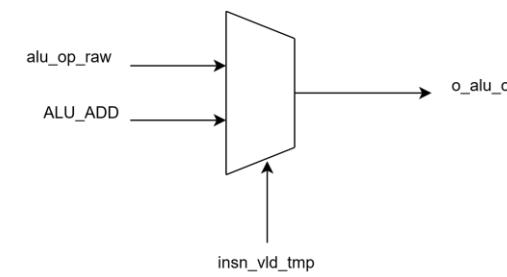
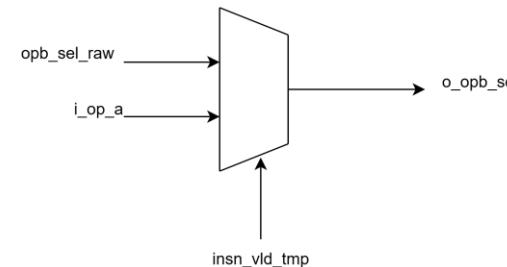
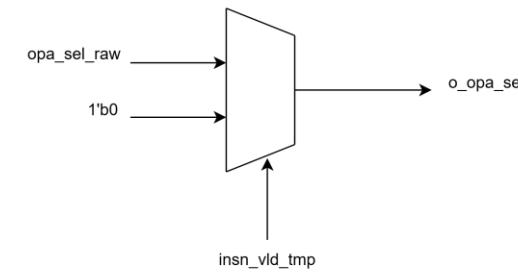
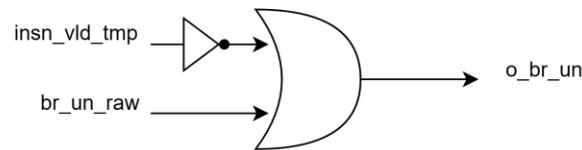
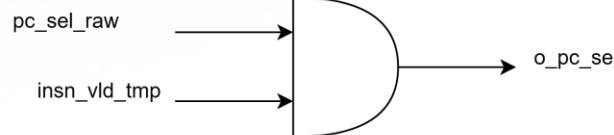
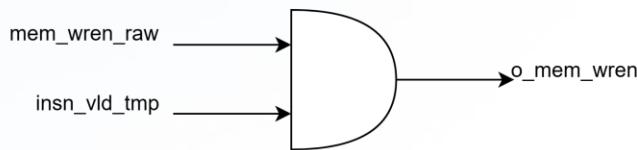
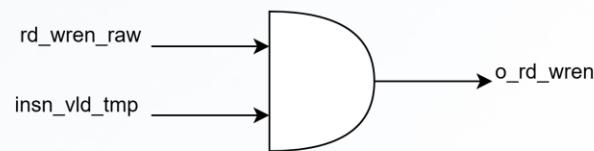
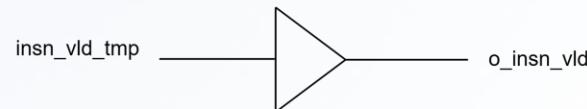
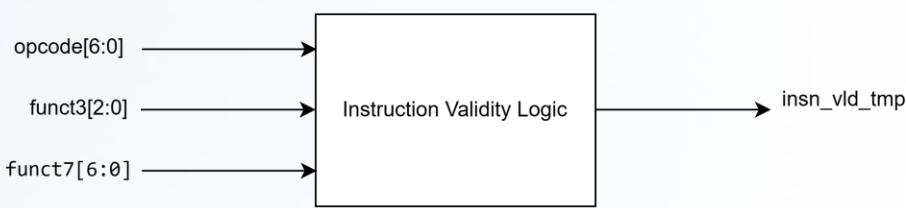
Sơ đồ khối của Control Unit



Sơ đồ khối của Control Unit



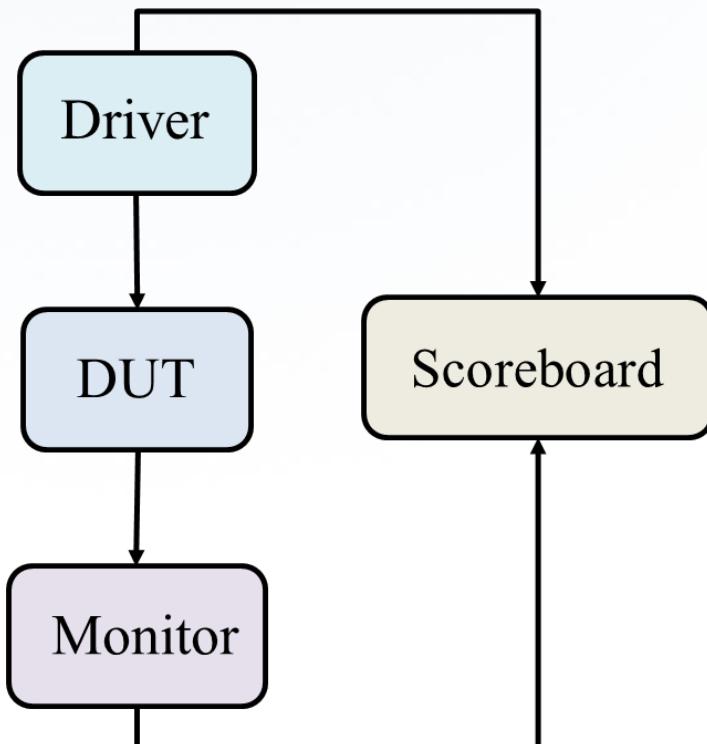
Sơ đồ khối của Control Unit



Verification Strategy

- **Mục tiêu:** Đảm bảo bộ xử lý hoạt động đúng chức năng với đầy đủ các nhóm lệnh RV32I (R, I, Load/Store, B, J, U).
- **Môi trường: Sử dụng Grand Test do TA cung cấp.**

Mô hình kiểm thử:



Thành phần	Vai trò
Driver	Cung cấp các chuỗi lệnh (theo dạng HEX) được đọc từ tập tin isa_4b.hex
DUT (Device Under Test)	Bộ xử lý RV32I đơn chu kì - Module pipelined
Monitor	Theo dõi tín hiệu o_pc_debug, o_insn_vld, và I/O memory-mapped
Scoreboard	So sánh kết quả thực tế với kết quả mong đợi, từ đó quyết định hiển thị PASS / ERROR cho từng test case.

Kết quả kiểm thử

Kết quả: Vượt qua toàn bộ bài kiểm thử

Model

1

```
===== Result =====
Total Clock Cycles Executed = 11132
Total Instructions Executed = 4831
Total Branch Instructions = 1604
Total Branch Mispredictions = 1452

-----
Instruction Per Cycle (IPC) = 0.43
Branch Misprediction Rate = 90.52 %

END of ISA tests
```

```
===== Result =====
Total Clock Cycles Executed = 7841
Total Instructions Executed = 4831
Total Branch Instructions = 1604
Total Branch Mispredictions = 1452

-----
Instruction Per Cycle (IPC) = 0.62
Branch Misprediction Rate = 90.52 %

END of ISA tests
```

Model

2

add.....	PASS
addi....	PASS
sub....	PASS
and....	PASS
andi....	PASS
or.....	PASS
ori....	PASS
xor....	PASS
xori....	PASS
slt....	PASS
slti....	PASS
sltu....	PASS
sltiu....	PASS
sll....	PASS
slli....	PASS
srl....	PASS
srlti....	PASS
sra....	PASS
srai....	PASS
lw.....	PASS
lh.....	PASS
lhu....	PASS
lb.....	PASS
lbu....	PASS
sw.....	PASS
sh.....	PASS
sb.....	PASS
auipc...	PASS
lui....	PASS
beq....	PASS
bne....	PASS
blt....	PASS
bltu....	PASS
bge....	PASS
bgeu....	PASS
jal....	PASS
jalr....	PASS
malign...	ERROR
iosw....	PASS

Application

- Chủ đề: **Convert a hexadecimal number to a decimal number and display on seven-segment LEDs**

Mục tiêu

- Nhận giá trị đầu vào từ Switches
- Hiển thị dạng **Decimal** lên **HEX0...HEX3**
- Hiển thị dạng **Hexadecimal** lên **HEX4...HEX7**
- Bật/tắt mỗi phần hiển thị bằng SW[14:13]

Cách hoạt động

- **Chuyển đổi Decimal:** Chia lặp cho 10 và ánh xạ lên các HEX
- **Chuyển đổi Hexadecimal:** Tách các nibble và ánh xạ lên các HEX

Demo trên kit DE2



Hình 15: *Bật cụm LED Decimal - tắt cụm Hexadecimal*

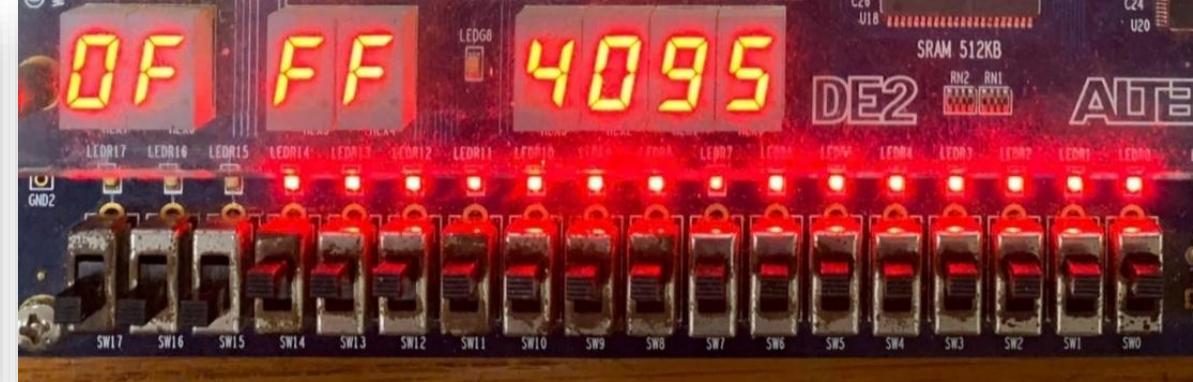


Hình 16: *Bật cụm LED Hexadecimal - tắt cụm Decimal*

Video demo trên FPGA: [KTMT_L02_7_Application.mp4](#)



Hình 17: *Bật hết cả 2 cụm Decimal và Hexadecimal*



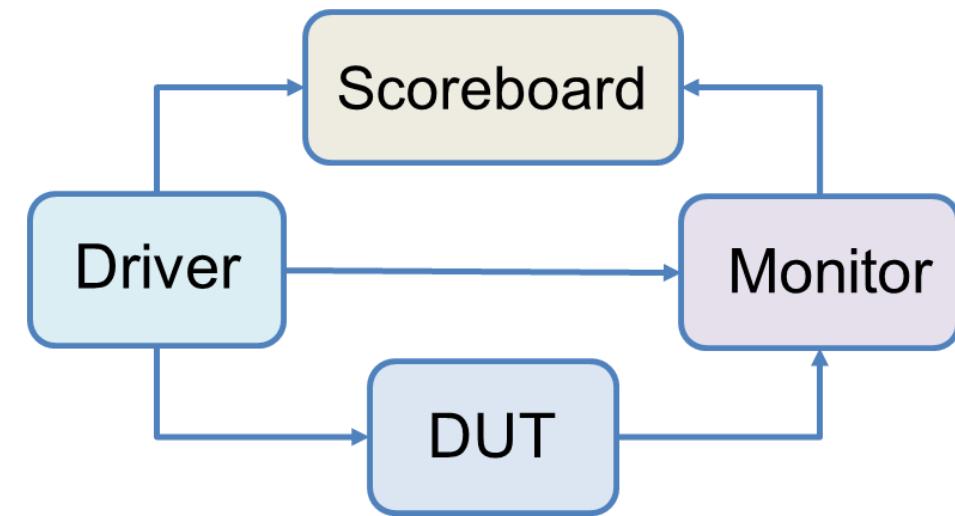
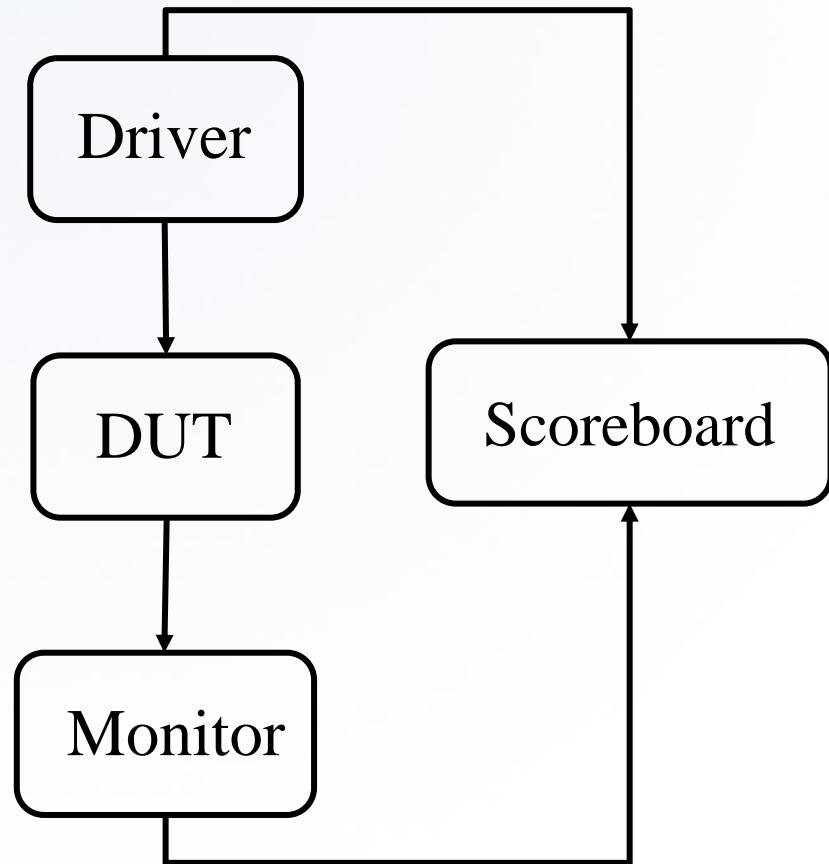
Hình 18: *Bật hết tất cả các Switches*

**Cảm ơn thầy và các bạn đã
lắng nghe**

Đánh giá & Q&A

- ✓ **Thành công:** Core single-cycle RV32I hoạt động đúng chức năng trong cả mô phỏng và chạy thực tế trên FPGA.
- ⚠ **Hạn chế:** Tài nguyên sử dụng còn cao (36% logic elements) do một số mô tả ở mức hành vi và cấu trúc MUX-tree.
- ✓ **Cải thiện:** Tối ưu hóa tài nguyên và cấu trúc mô tả RTL trong các thiết kế tương lai.

Cảm ơn thầy và các bạn đã lắng nghe!



Ứng dụng: Demo trên Kit DE2

Chủ đề: "Chuyển đổi số Hexa sang Decimal và hiển thị trên 7-segment LEDs."



Bật cụm Decimal (SW[13]=1)



Bật cụm Hexadecimal (SW[14]=1)



Bật tất cả Switches (Giá trị 0FFF)