

---

Nhóm 1 - CE118.Q11.VMTN

# THIẾT KẾ VI XỬ LÝ 16-BIT

theo kiến trúc tập lệnh đơn giản

GV hướng dẫn: Tạ Trí Đức

---

# DANH SÁCH THÀNH VIÊN

**24520344** Nguyễn Việt Dũng (Nhóm trưởng)

**24520301** Tạ Vương Điền

**24521177** Huỳnh Hữu Nguyễn

**24520839** Phan Công Đăng Khoa

**24520816** Lê Đăng Khoa



# -MỤC LỤC-

01

---

TỔNG QUAN  
THIẾT KẾ

02

---

PIPELINE &  
MẠCH NHÂN

03

---

KIỂM ĐỊNH

# TẬP LỆNH

# 01

## TỔNG QUAN THIẾT KẾ

### • RRR:

Ins	Opcode 15 13	Rs 12 10	Rt 9 7	Rd 6 4	Funct 3 0	Assembly-Code Format	Meaning
add	000	Rs	Rt	Rd	0000	add Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} + \text{Reg[Rt]}$
inc	000	Rs	000	Rd	0001	inc Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} - \text{Reg[Rt]}$
sub	000	Rs	Rt	Rd	0010	sub Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} + 1$
dec	000	Rs	000	Rd	0011	dec Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} - 1$
and	000	Rs	Rt	Rd	0100	and Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \& \text{Reg[Rt]}$
or	000	Rs	Rt	Rd	0101	or Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]}   \text{Reg[Rt]}$
nand	000	Rs	Rt	Rd	0110	nand Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \sim \& \text{Reg[Rt]}$
xor	000	Rs	Rt	Rd	0111	xor Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \wedge \text{Reg[Rt]}$
shfl	000	Rs	000	Rd	1000	shfl Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \ll 1$
shfr	000	Rs	000	Rd	1001	shfr Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \gg 1$
sll	000	Rs	000	Rd	1010	shll Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \ll 2$
srr	000	Rs	000	Rd	1011	shrr Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \gg 2$
sra	000	Rs	000	Rd	1100	shra Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \gg 1$ (giữ bit dấu)
mul	000	Rs	Rt	Rd	1101	mul Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \times \text{Reg[Rt]}$
slt	000	Rs	Rt	Rd	1110	slt Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow (\text{Reg[Rs]} < \text{Reg[Rt]})$ ? 1 : 0
sgt	000	Rs	Rt	Rd	1111	sgt Rd, Rs, Rt	$\text{Reg[Rd]} \leftarrow (\text{Reg[Rs]} > \text{Reg[Rt]})$ ? 1 : 0

### • RI:

Ins	Opcode 15 13	Rs 12 10	Imm 9 0	Assembly-Code Format	Meaning
lw	100	Rs	Imm	lw Rs, Imm	$\text{Reg[Rs]} \leftarrow \text{Mem[Imm]}$
sw	101	Rs	Imm	sw Rs, Imm	$\text{Mem[Imm]} \leftarrow \text{Reg[Rs]}$
jnz	110	Rs	Imm	jnz Rs, Imm	$\text{PC} \leftarrow (\text{Reg[Rs]} \neq 0) ? \text{Imm} : \text{PC} + 1$
li	111	Rs	Imm	li Rs, Imm	$\text{Reg[Rs]} \leftarrow \text{Imm}$

### • Extend (Mã giả):

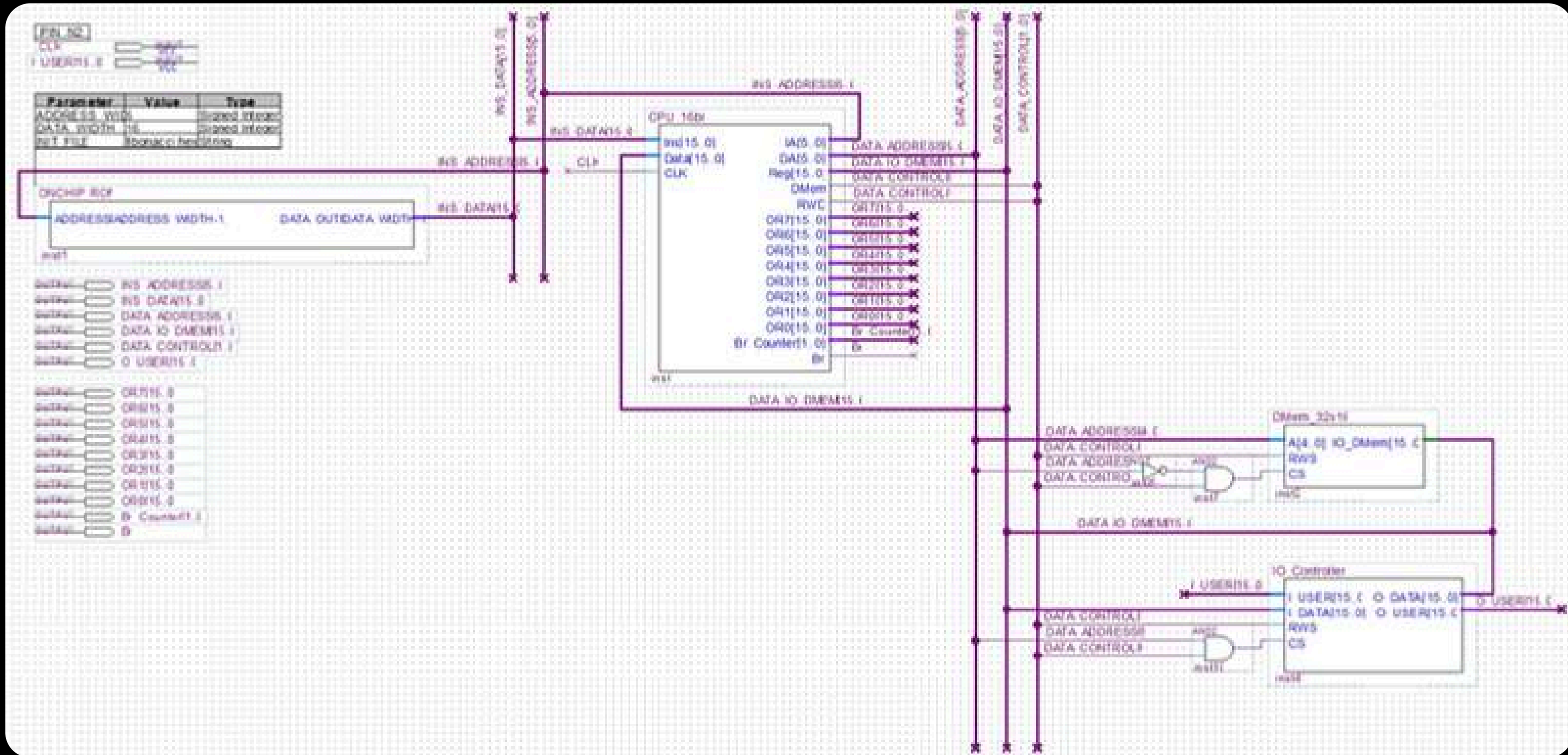
Ins	Opcode 15 13	Rs 12 10	Rt 9 7	Rd 6 4	Funct 3 0	Assembly-Code Format	Meaning
not	000	Rs	Rs	Rd	1000	not Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \sim \& \text{Reg[Rs]}$
mov	000	Rs	Rs	Rd	0110	mov Rd, Rs	$\text{Reg[Rd]} \leftarrow \text{Reg[Rs]} \& \text{Reg[Rs]}$
nop	001	Rs	Rs	0000000		nop	$\text{Reg[Rs]} \leftarrow \text{Reg[Rs]} + 0$
jmp	011	Rs	Rs	Imm		jmp Imm	$\text{PC} \leftarrow (\text{Reg[Rs]} == \text{Reg[Rs]}) ?$ Imm : PC + 1

### • RRI:

Ins	Opcode 15 13	Rs 12 10	Rt 9 7	Imm 6 0	Assembly-Code Format	Meaning
addi	001	Rs	Rt	Imm	addi Rs, Rt, Imm	$\text{Reg[Rt]} \leftarrow \text{Reg[Rs]} + \text{SignExtImm}$
subi	010	Rs	Rt	Imm	subi Rs, Rt, Imm	$\text{Reg[Rt]} \leftarrow \text{Reg[Rs]} - \text{SignExtImm}$
beq	011	Rs	Rt	Imm	beq Rs, Rt, Imm	$\text{PC} \leftarrow (\text{Reg[Rs]} == \text{Reg[Rt]}) ? \text{Imm}$ : PC + 1



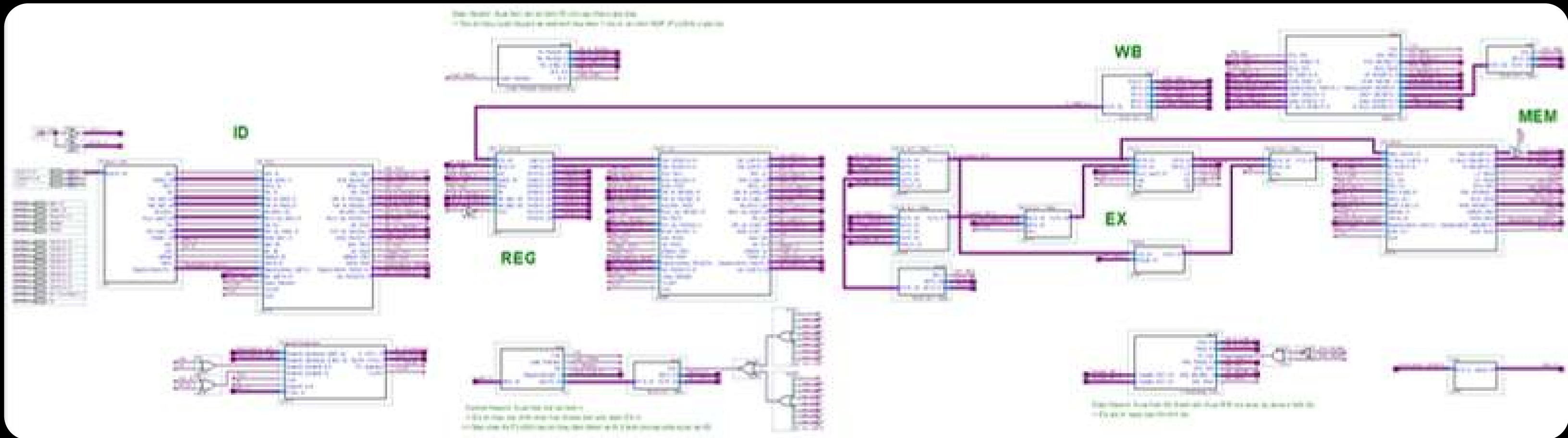
# TỔNG QUAN THIẾT KẾ



# DATAPATH – CPU 16-BIT

01

TỔNG QUAN THIẾT KẾ





02

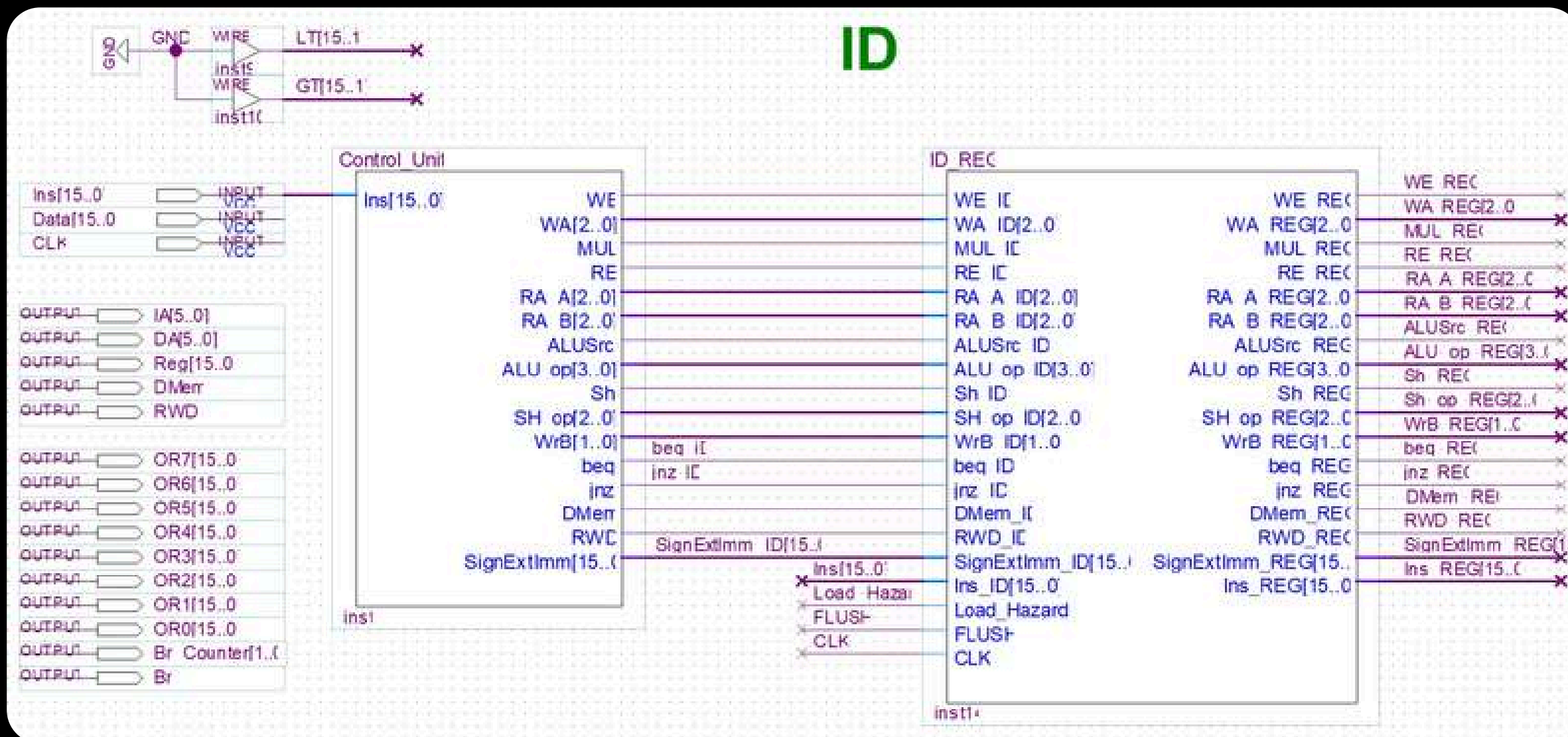
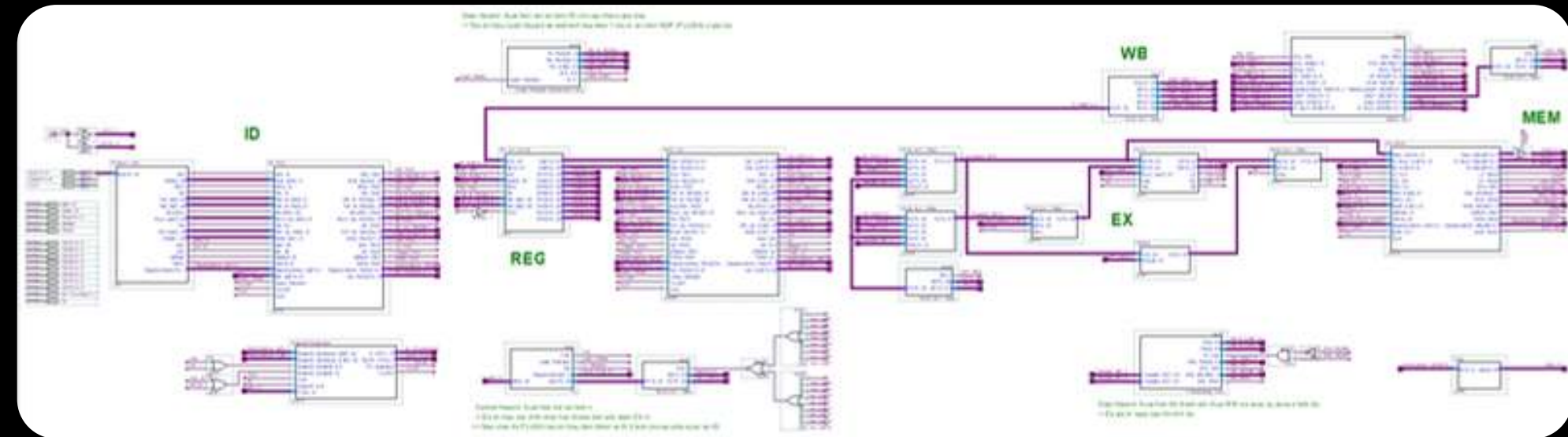
---

**PIPELINE &  
MẠCH NHÂN**

# PIPELINE - IF & ID

02

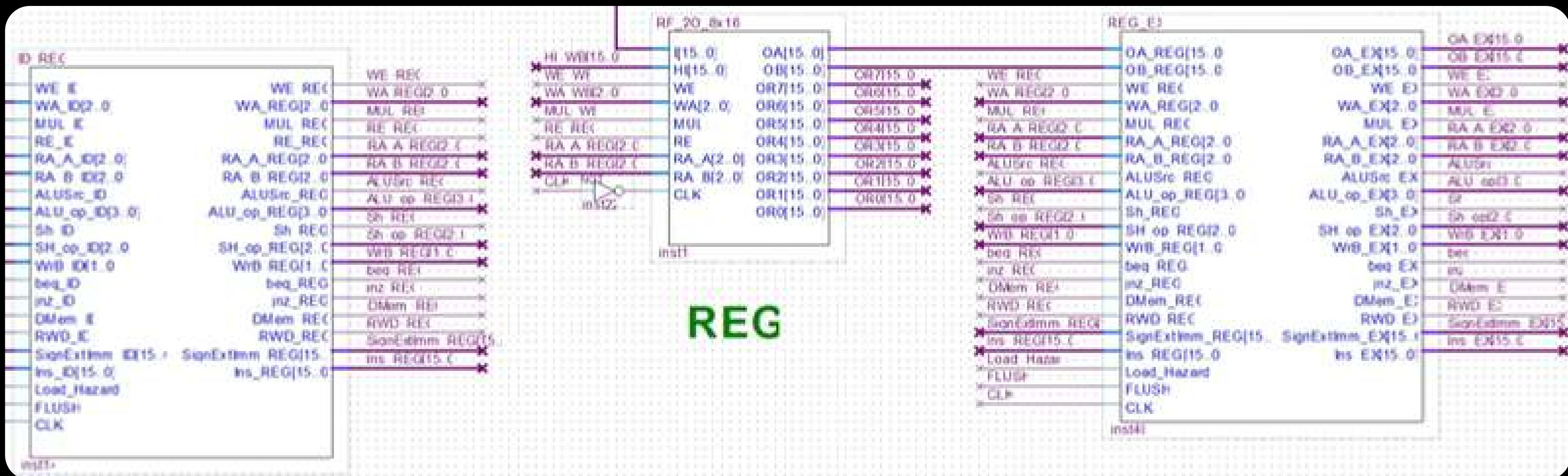
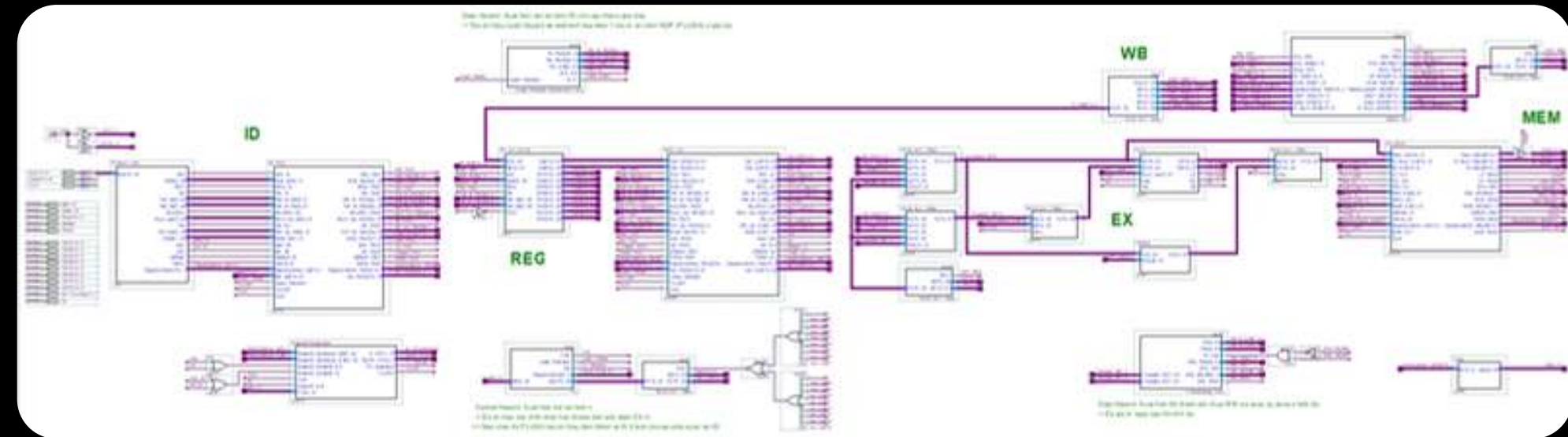
PIPELINE & MẠCH NHẬN



# PIPELINE - REG

02

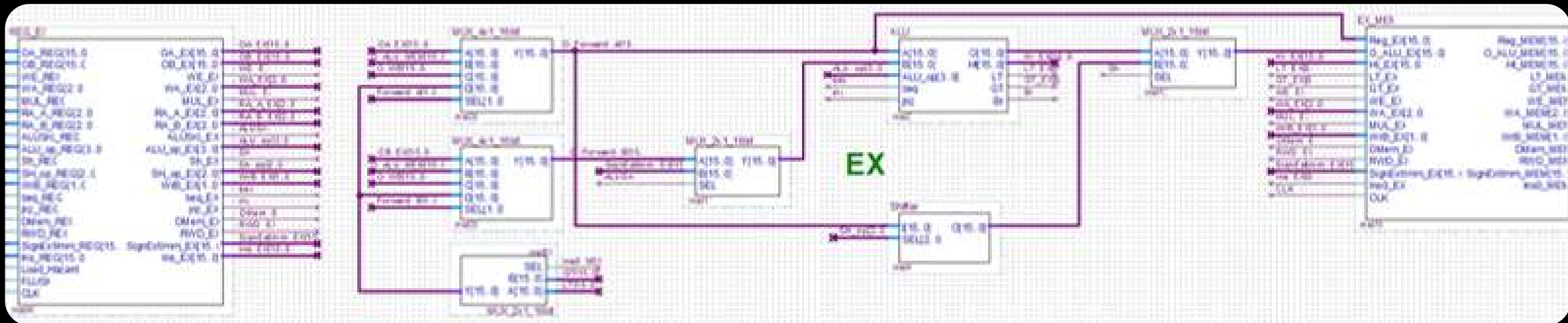
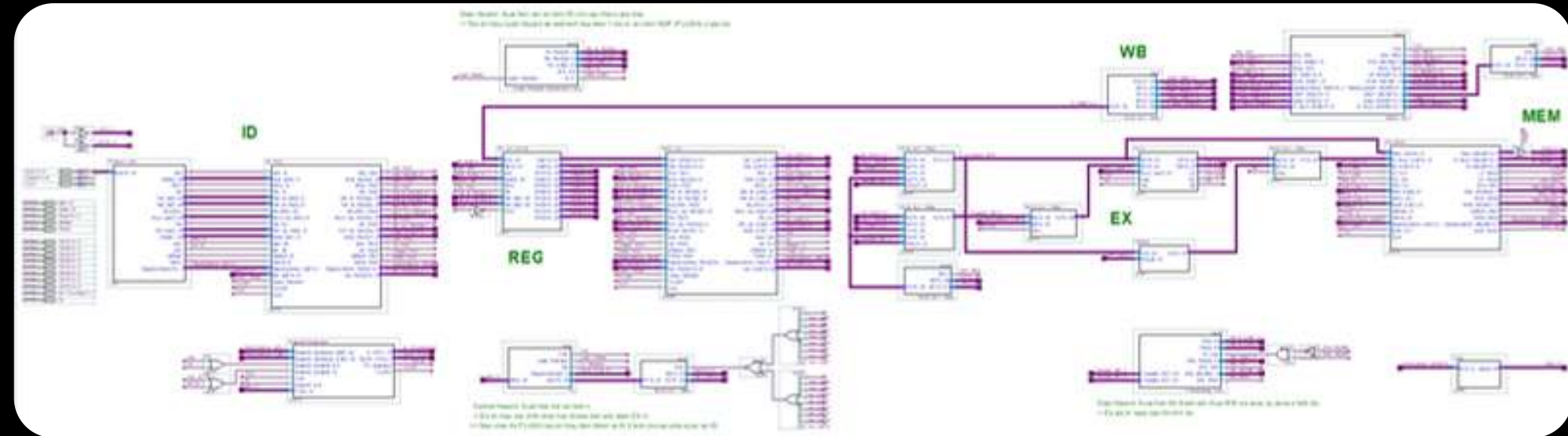
PIPELINE & MẠCH NHẬN



# PIPELINE - EX

02

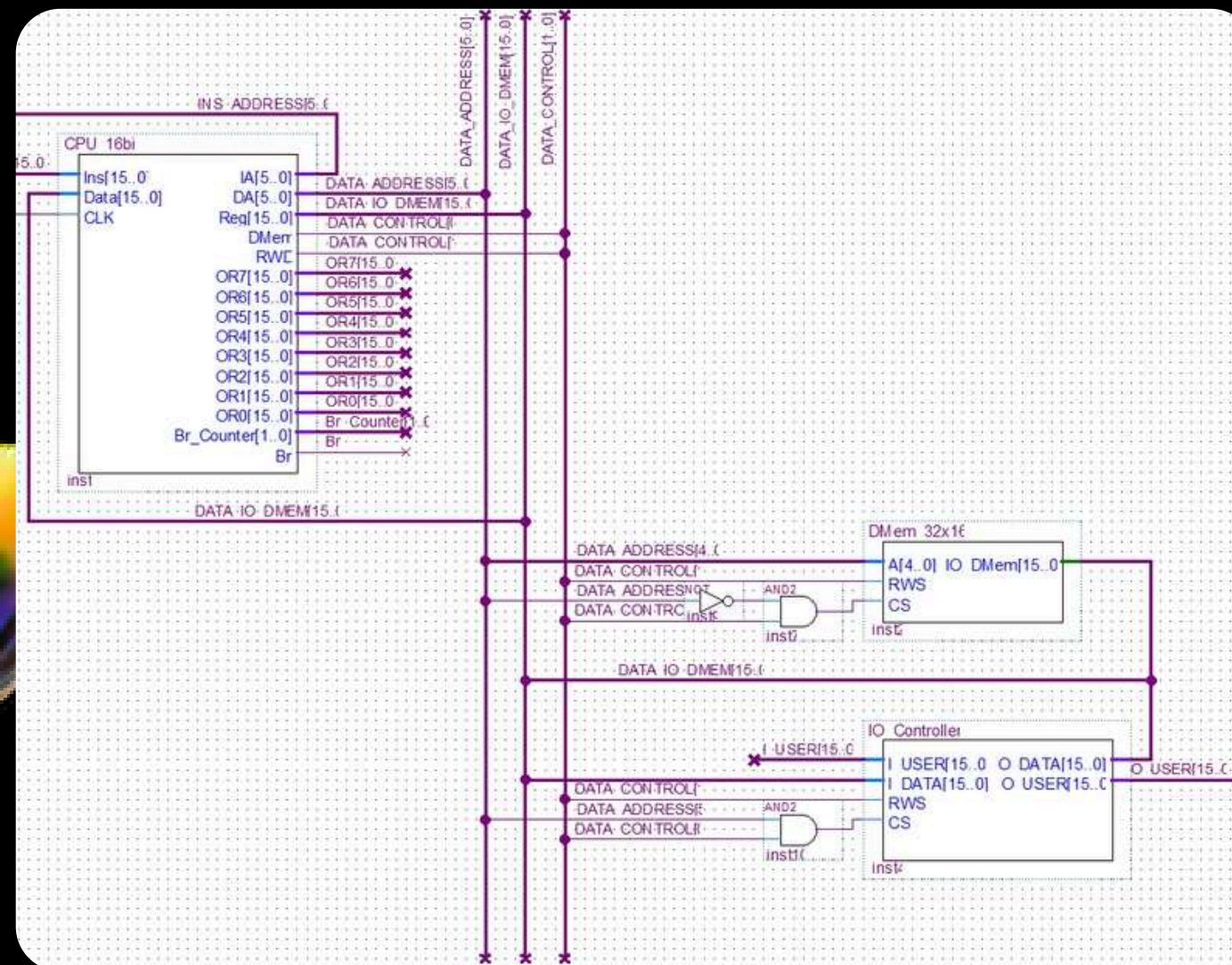
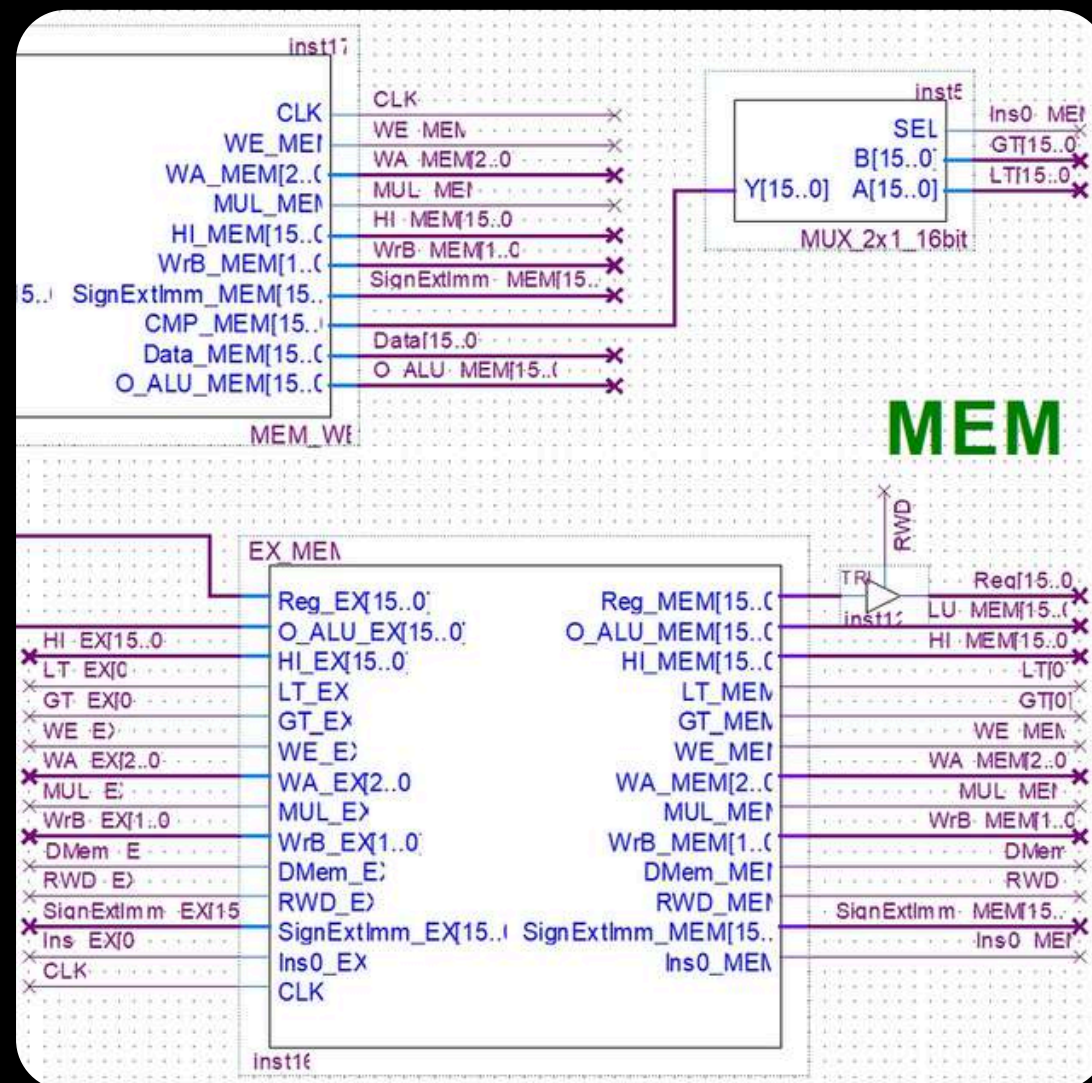
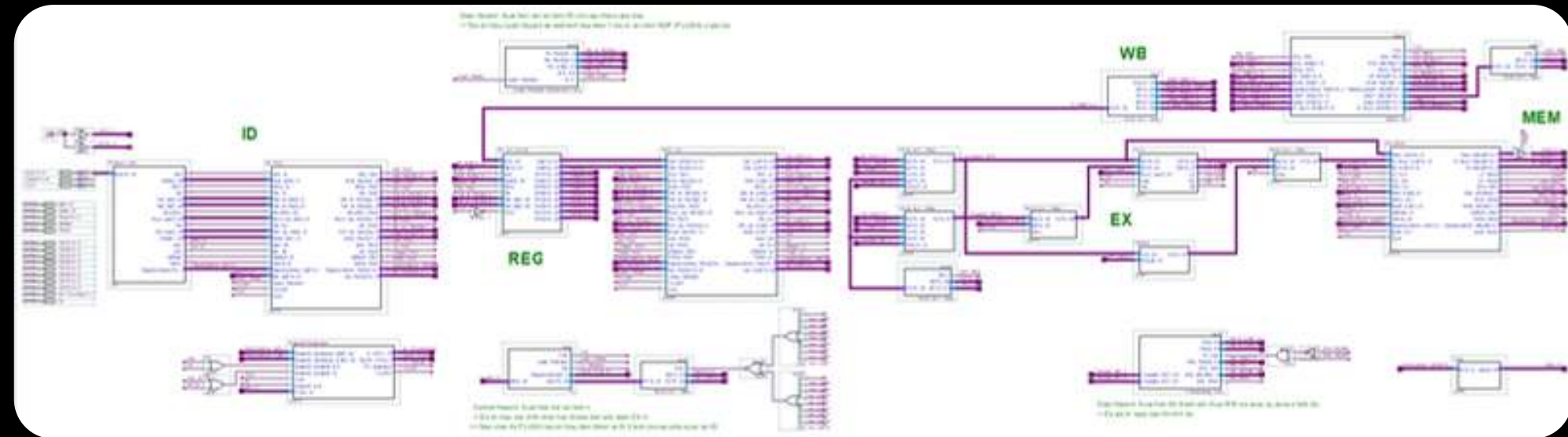
## PIPELINE & MẠCH NHẬN



# PIPELINE – MEM

02

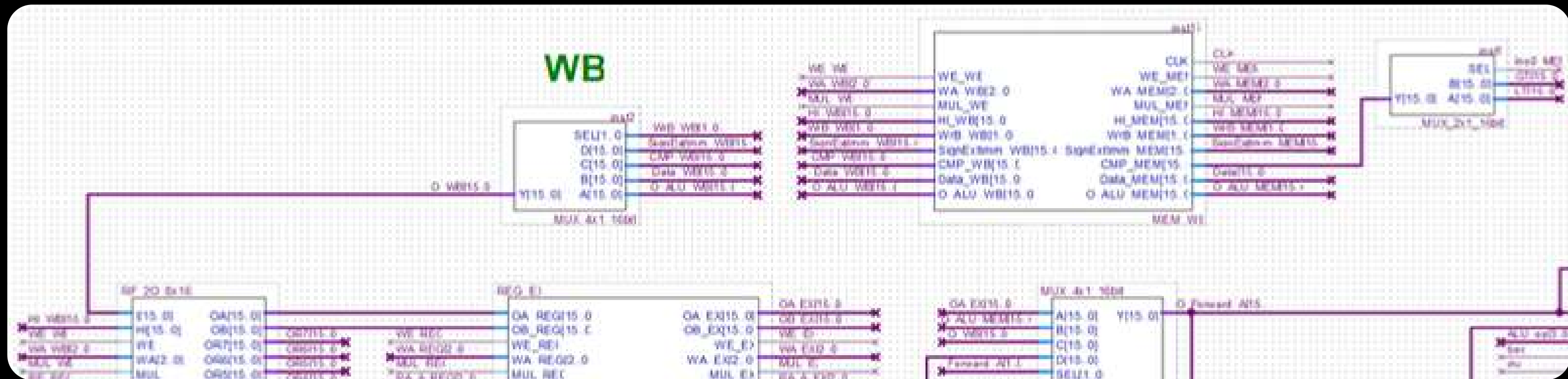
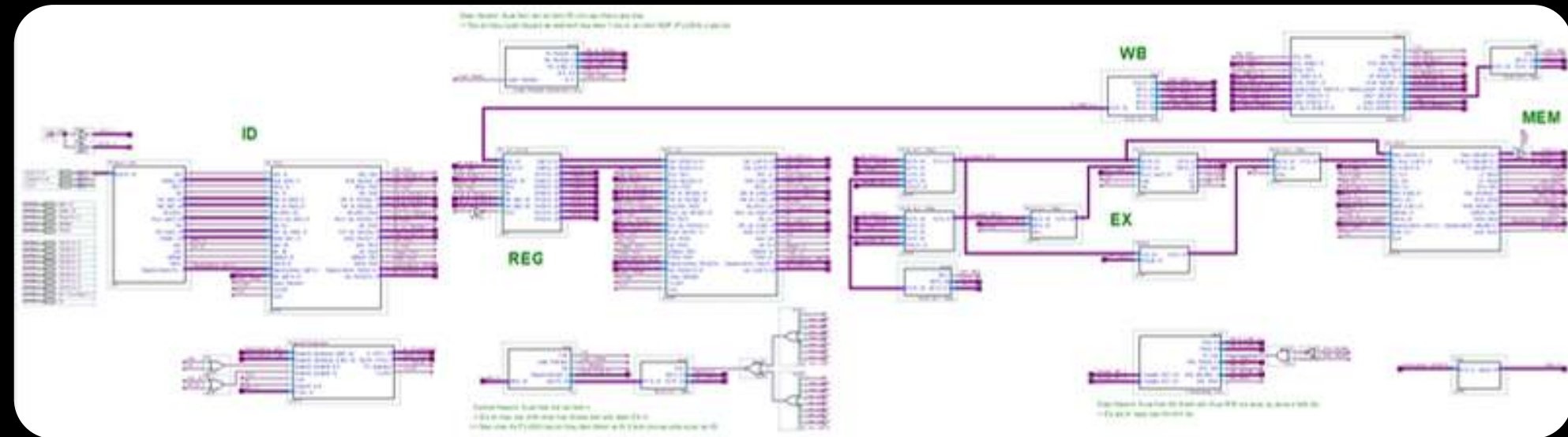
PIPELINE & MẠCH NHẬN



# PIPELINE - WB

02

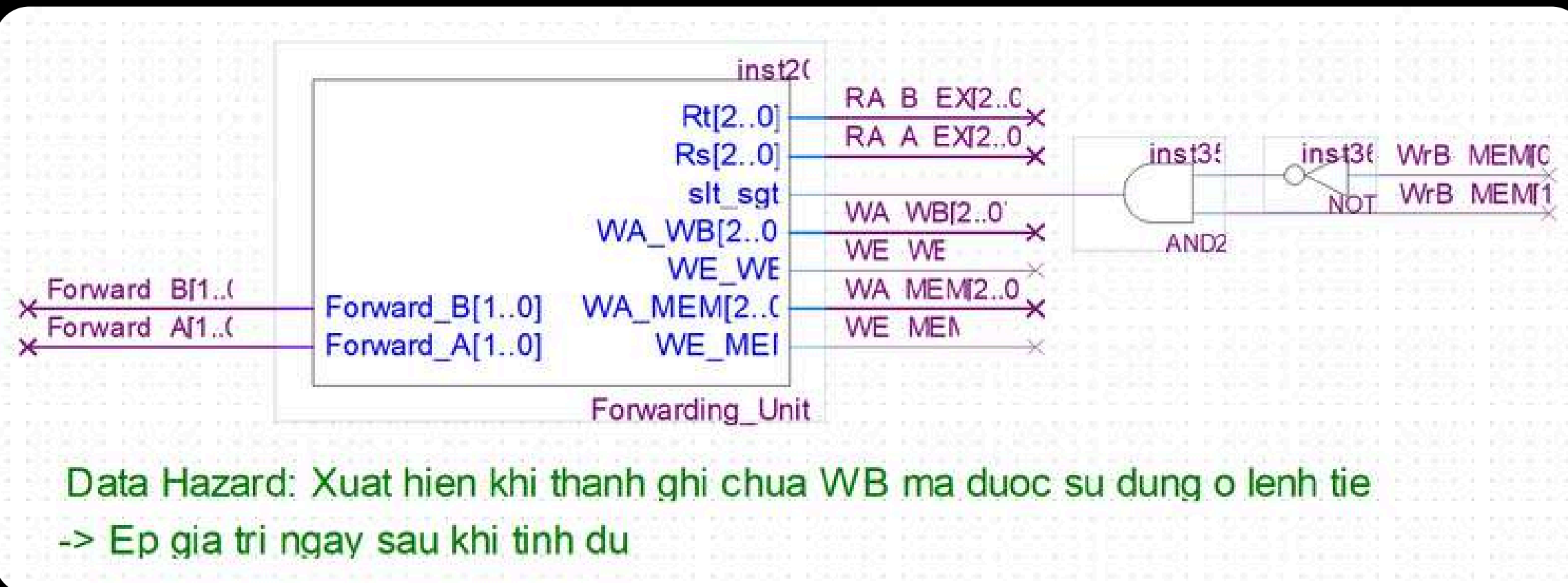
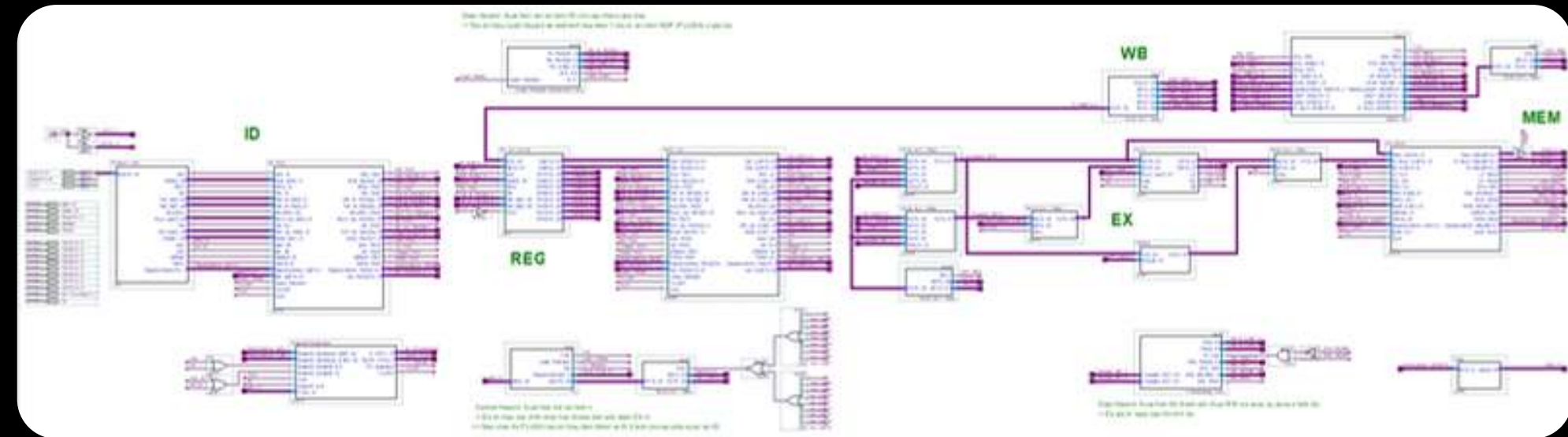
PIPELINE & MẠCH NHẬN



# PIPELINE – FORWARDING UNIT

02

PIPELINE & MẠCH NHẬN

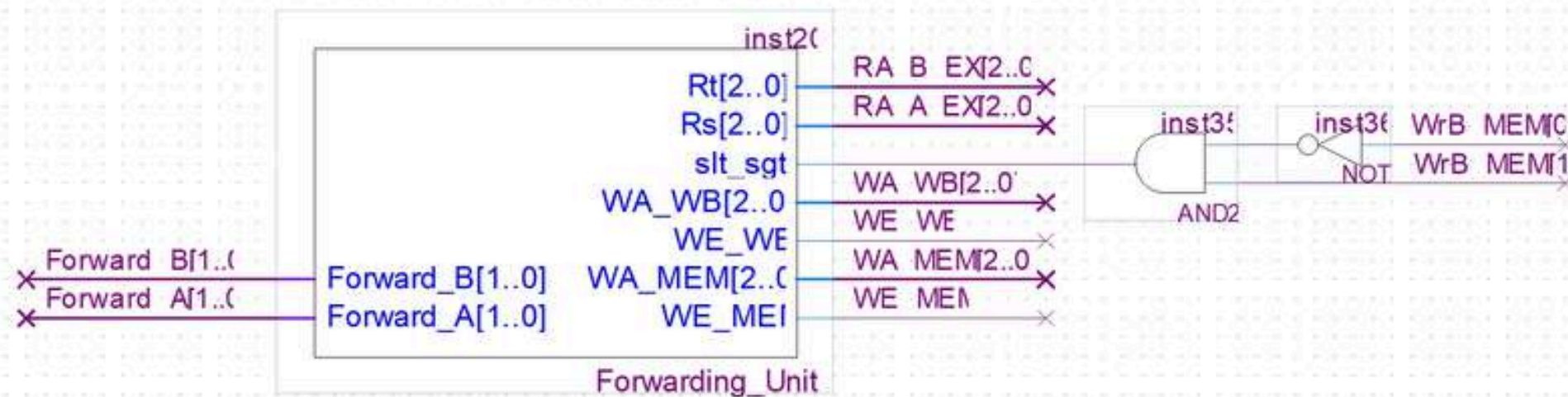


# PIPELINE – FORWARDING UNIT

02

PIPELINE & MẠCH NHÂN

```
sub    $2,    $1,    $3
and    $12,   $5,    $2    # 1b
or     $13,   $2,    $6    # 2a
add    $14,   $2,    $2    # No hazard
sw     $15,   100($2)    # No hazard
```



Data Hazard: Xuất hiện khi thành ghi chưa WB mà được sử dụng ở lệnh tiếp  
-> Ép giá trị ngay sau khi tính đủ

```
lw     $2,    20($1)
```

**nop**

```
or     $8,    $2,    $6
```

```
add    $9,    $4,    $2
```

```
slt    $1,    $6,    $7
```

```
add    $1,    $1,    $2
```

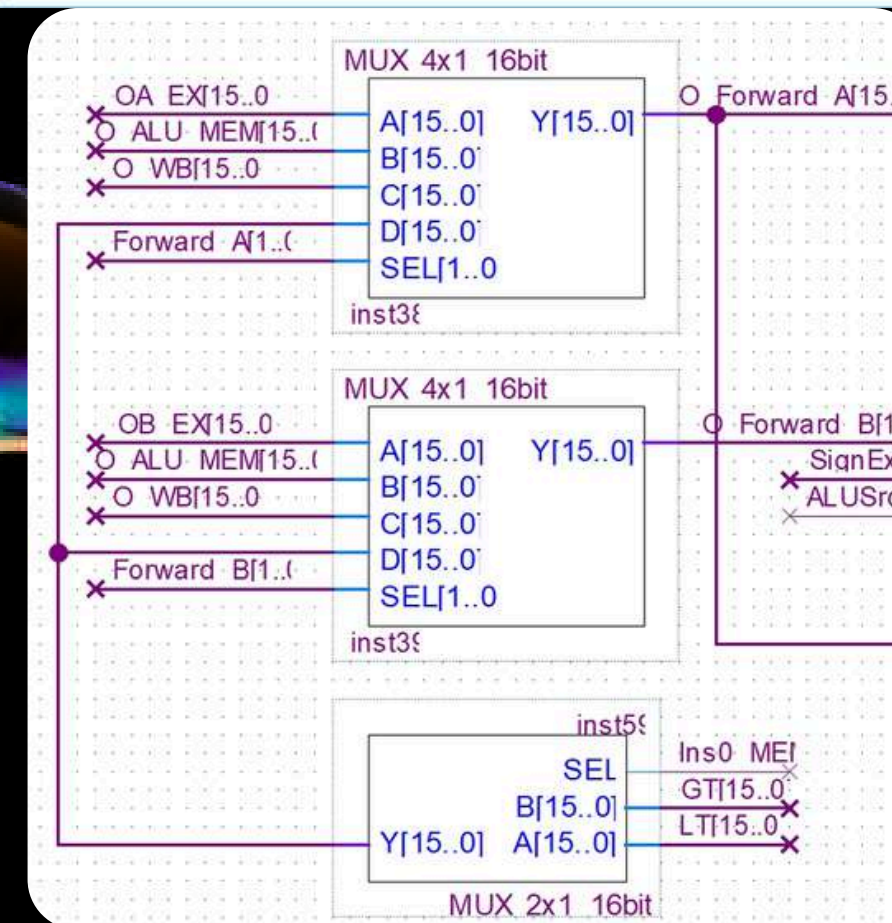
```
add    $1,    $1,    $3
```

```
add    $1,    $1,    $4
```

# 02

# PIPELINE & MẠCH NHÂN

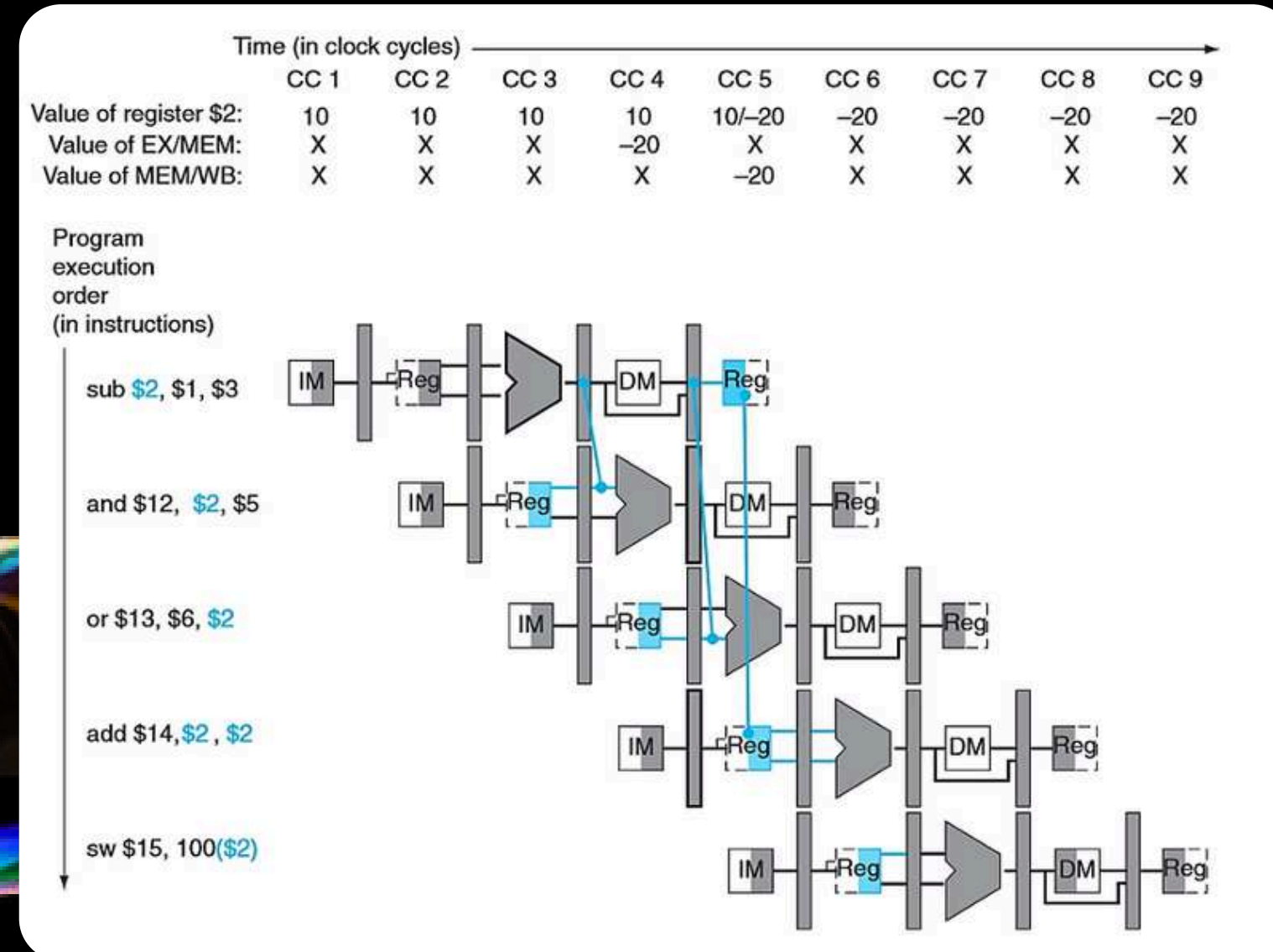
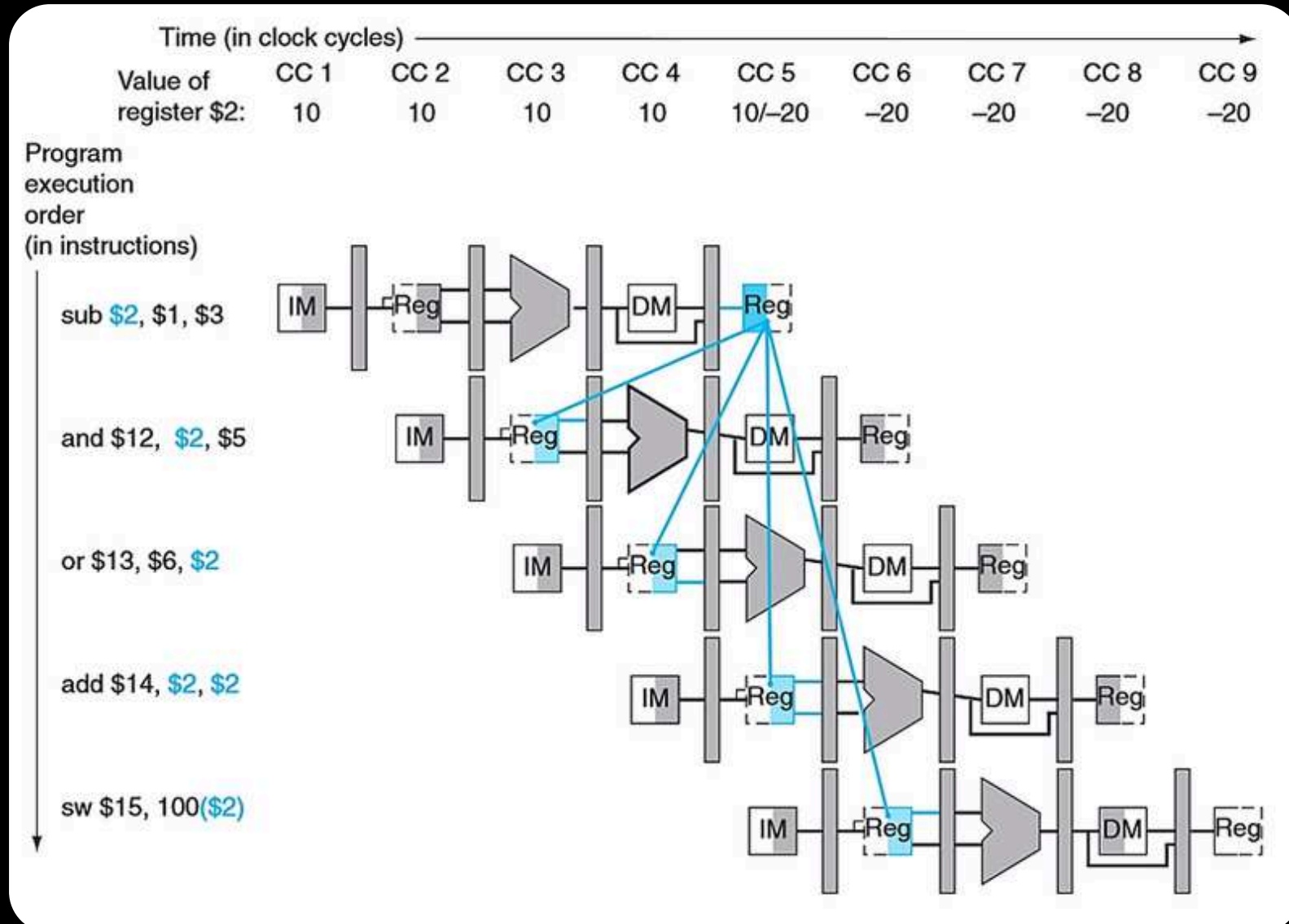
The diagram illustrates the internal logic of the ALU, showing the flow of data from inputs through various logic gates and multiplexers to produce the final output. The inputs are categorized into Forward A, Forward B, and Forward A/B. The Forward A section shows the calculation of Forward A[1:0] and Forward A[1:1]. The Forward B section shows the calculation of Forward B[1:0] and Forward B[1:1]. The Forward A/B section shows the calculation of Forward A/B[1:0] and Forward A/B[1:1]. The diagram includes various logic gates (AND, OR, XOR, XNOR) and multiplexers (MUX) that select between different inputs based on control signals. The inputs are labeled with their bit positions and the control signals are labeled with their bit positions. The outputs are labeled with their bit positions and the control signals.



# PIPELINE – FORWARDING UNIT

02

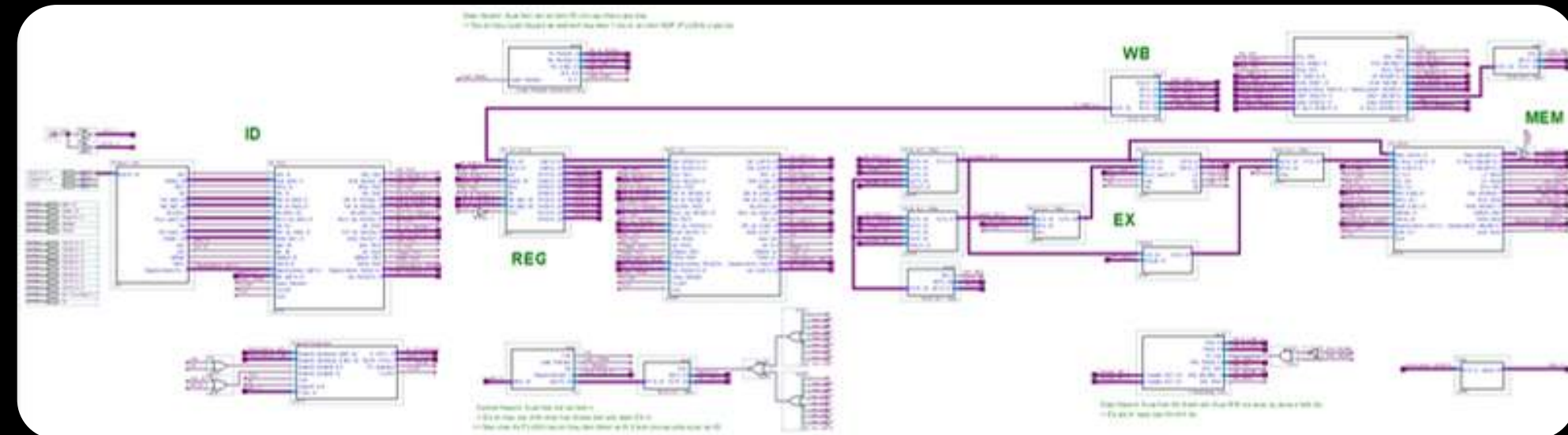
PIPELINE & MẠCH NHÂN



# PIPELINE – HAZARD DETECTION UNIT

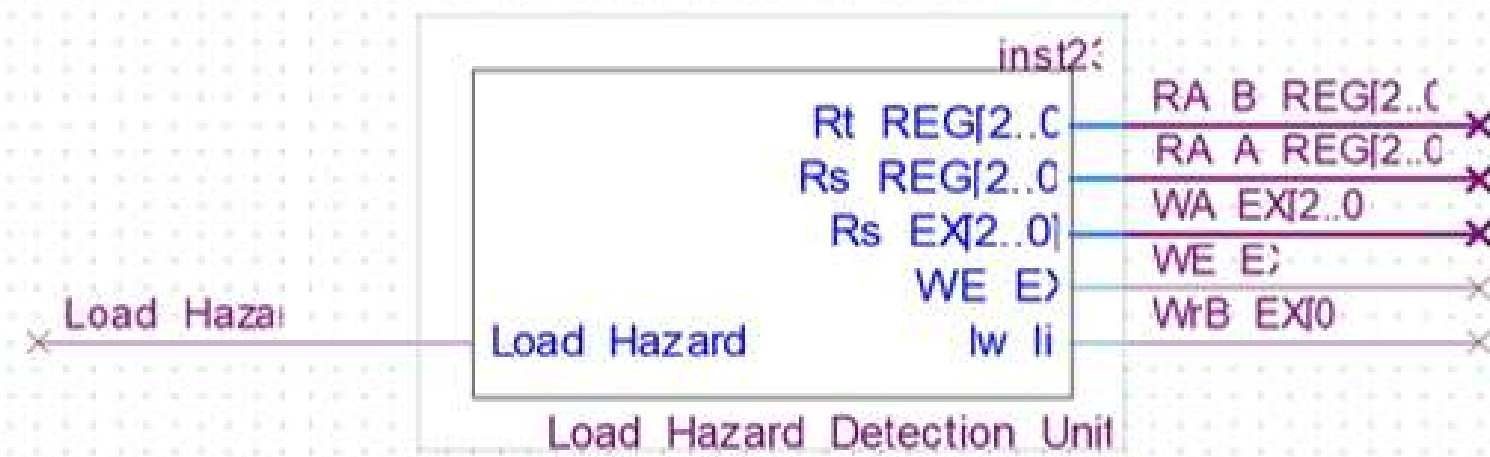
02

PIPELINE & MẠCH NHẬN



Data Hazard: Xuất hiện đối với lệnh RI chỉ cập nhật ở giai đoạn

-> Tạo tín hiệu Load Hazard để stall lệnh tiếp theo 1 chu kỳ và chen NOP (FLUSH) ở

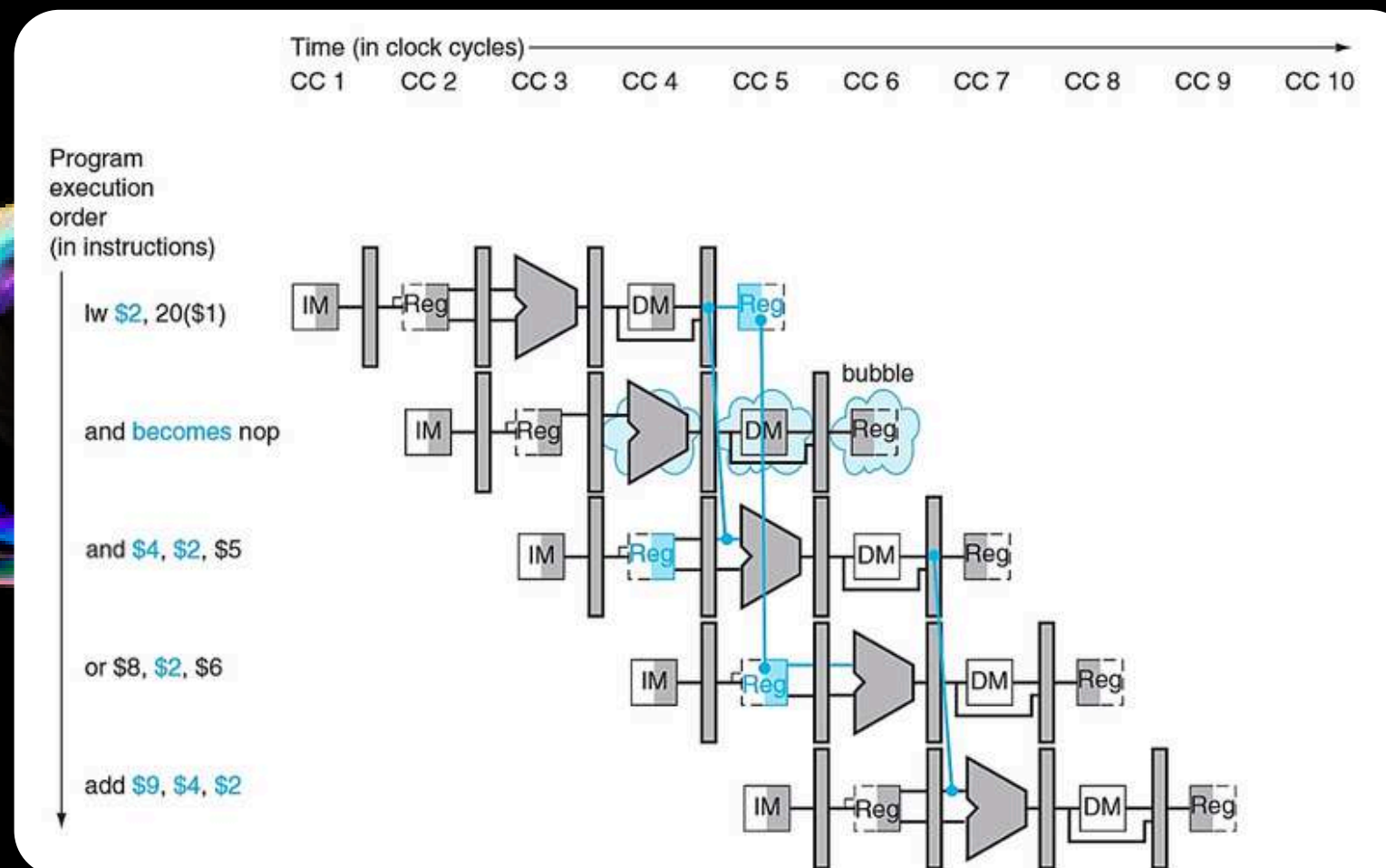
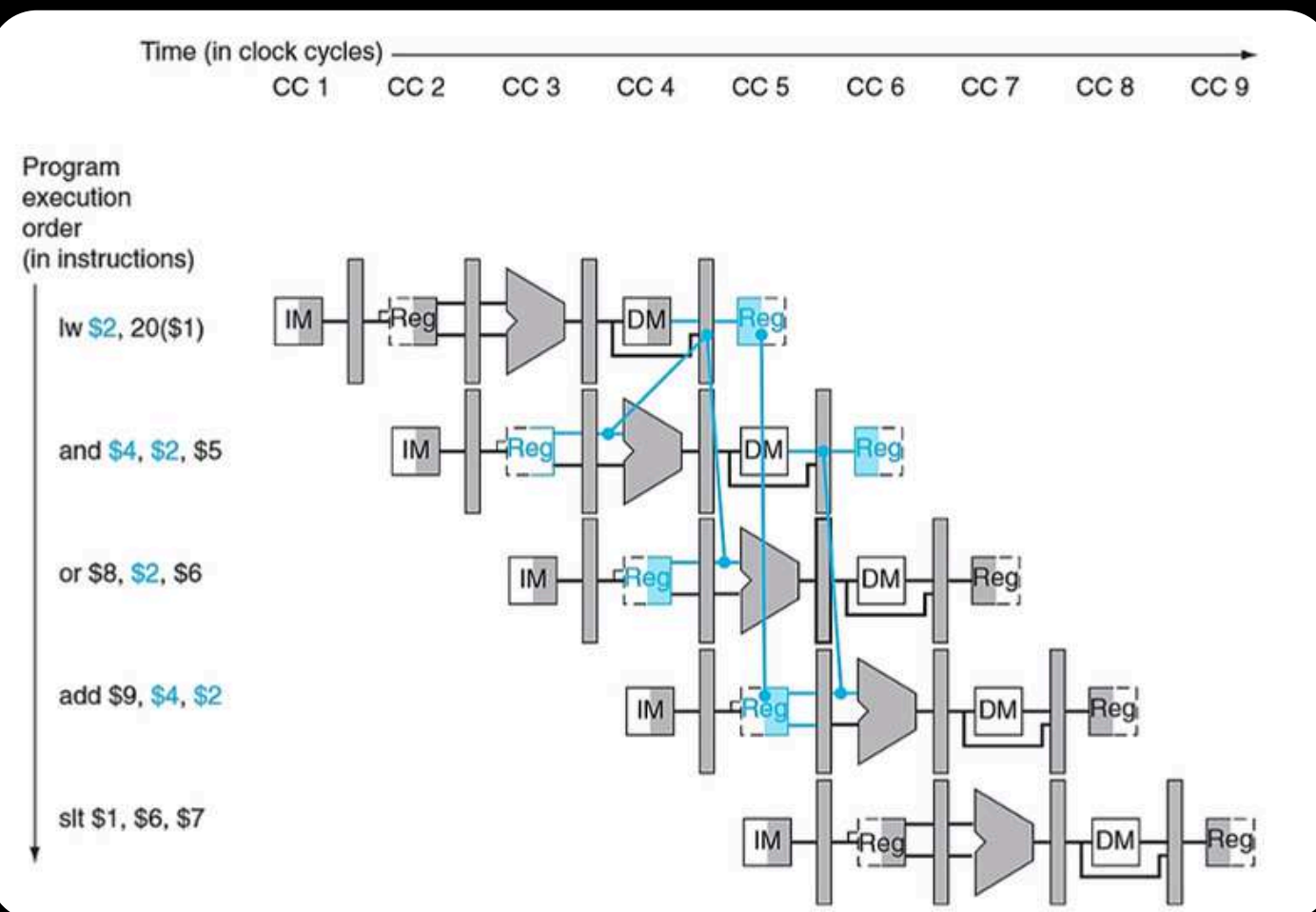
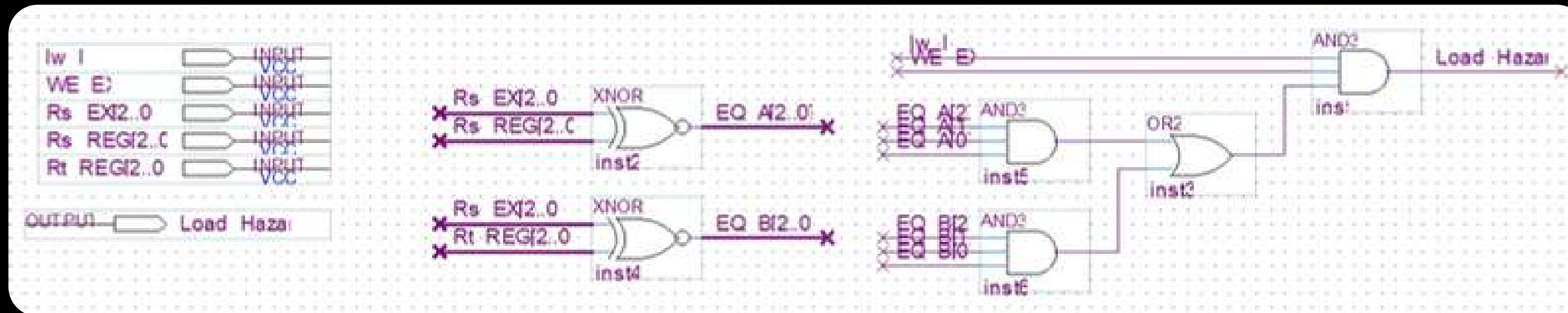


```
lw    $2, 20($1)
and   $4, $2, $5
or    $8, $2, $6
add   $9, $4, $2
slt   $1, $6, $7
```

# PIPELINE – HAZARD DETECTION UNIT

02

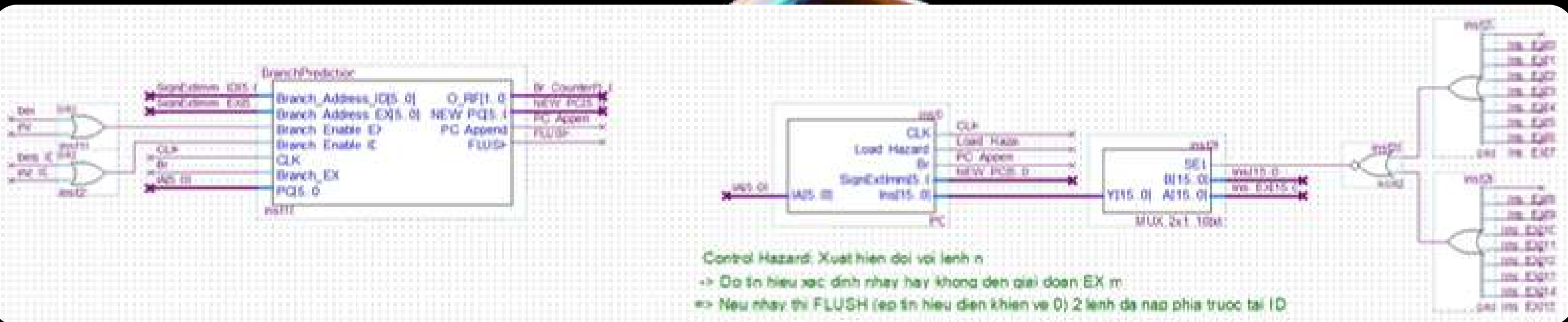
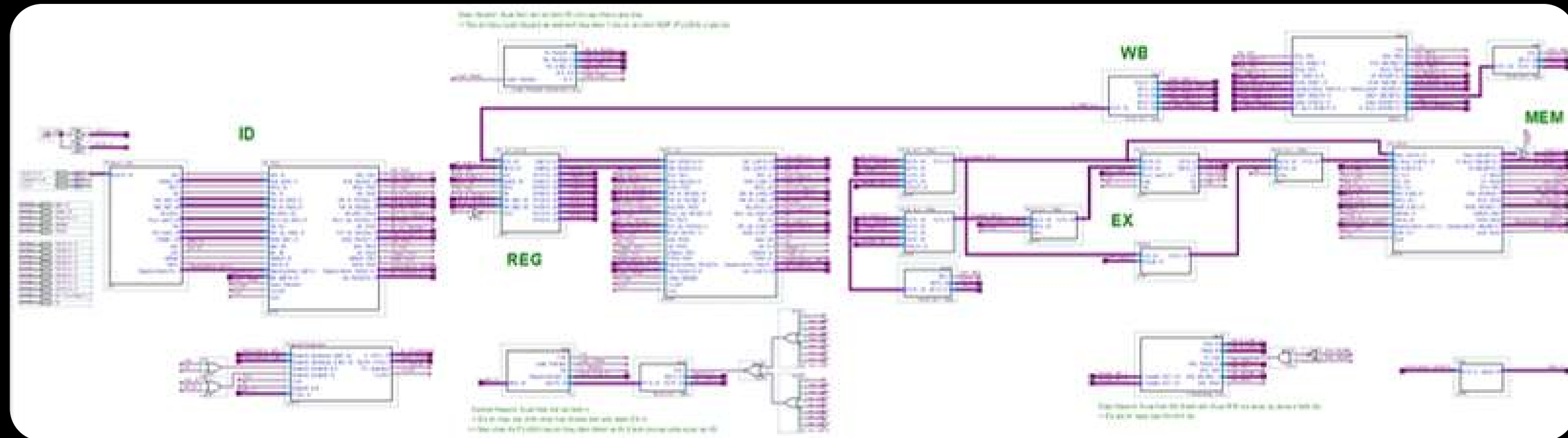
PIPELINE & MẠCH NHÂN



# PIPELINE – PC + BRANCH PREDICTION

02

PIPELINE & MẠCH NHẬN

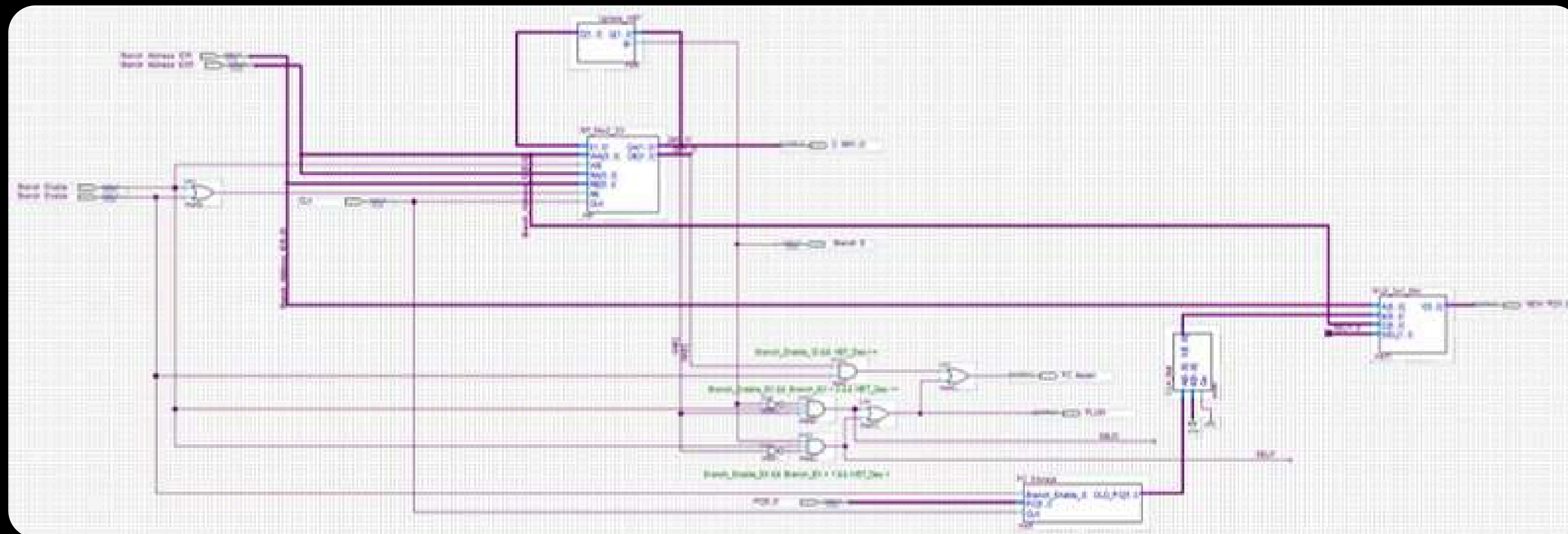
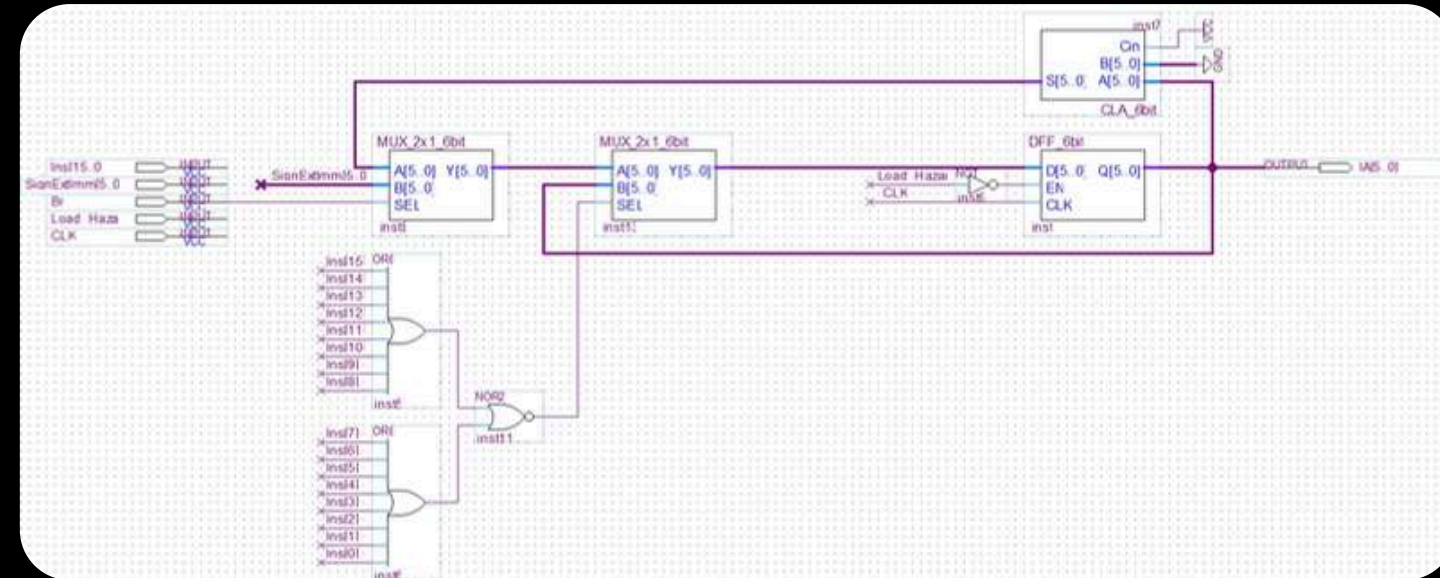


# PIPELINE – PC + BRANCH PREDICTION

02

```
int i = 0;
do{
    /* loop body */
    i = i + 1
}while(i < 10);

L1:    add    $t0, $0, $0
      /* loop body*/
      addi   $t0, $t0, 1
      slti   $t1, $t0, 10
      bne    $t1, $0, L1
```

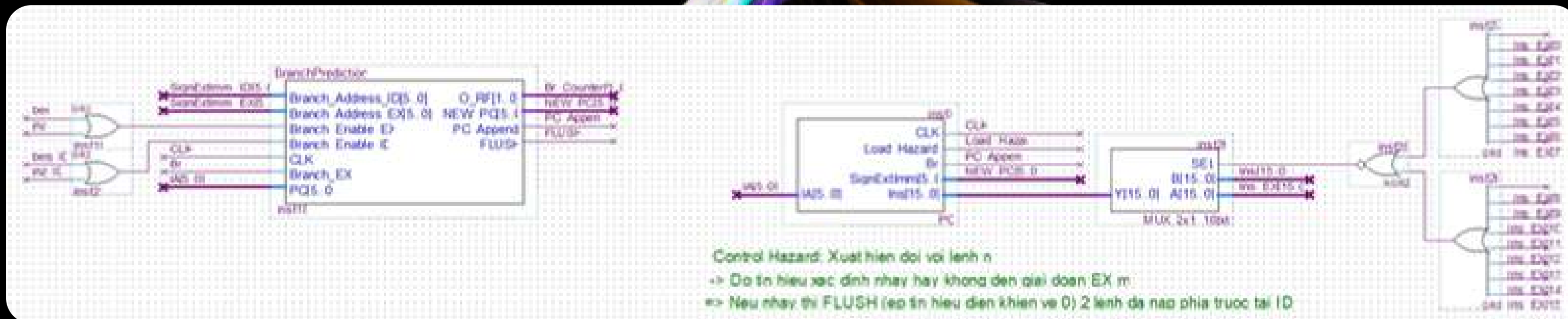
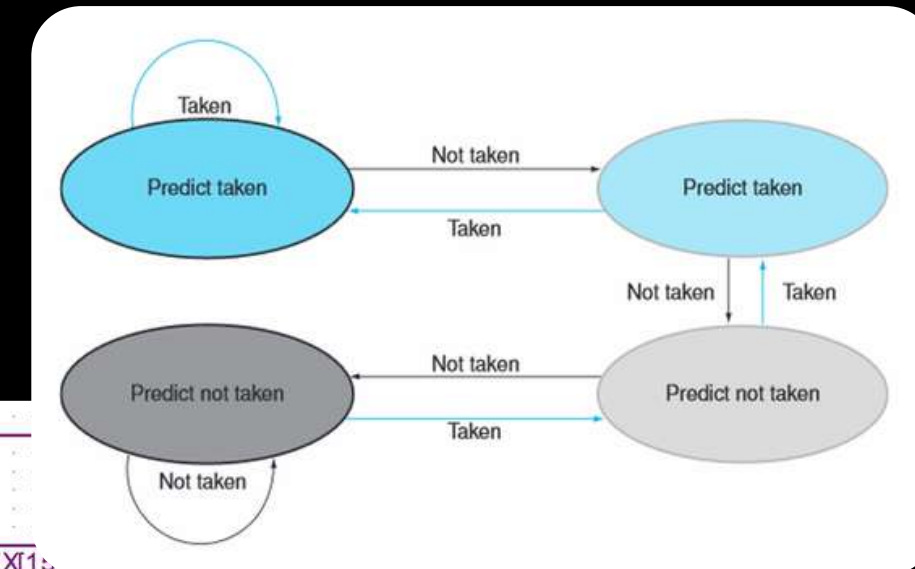
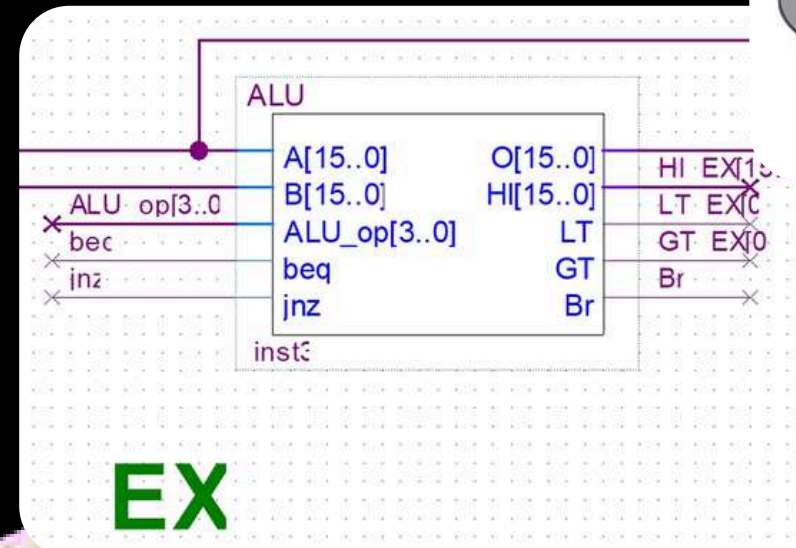
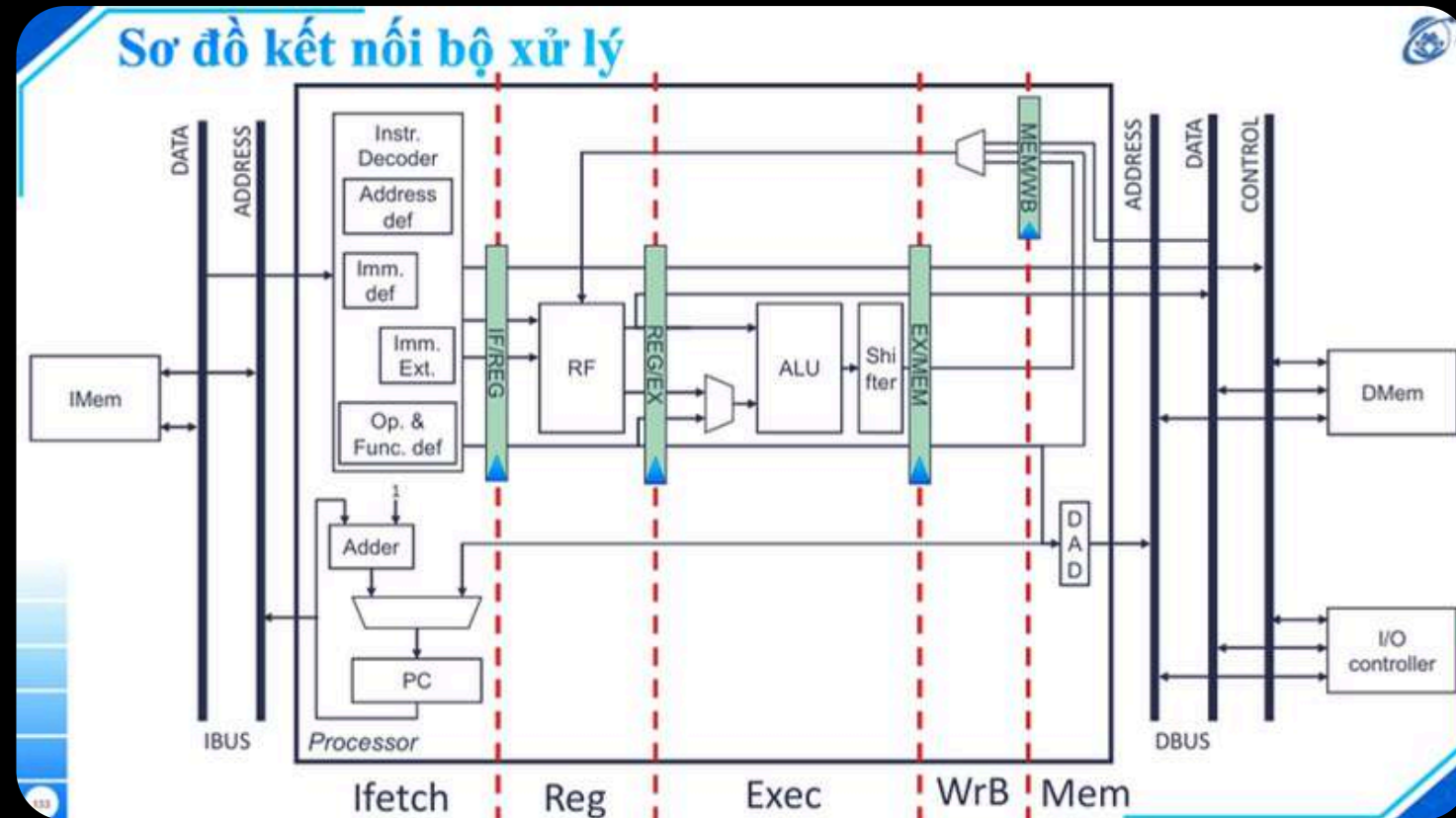


PIPELINE & MẠCH NHẬN

# PIPELINE – PC + BRANCH PREDICTION

02

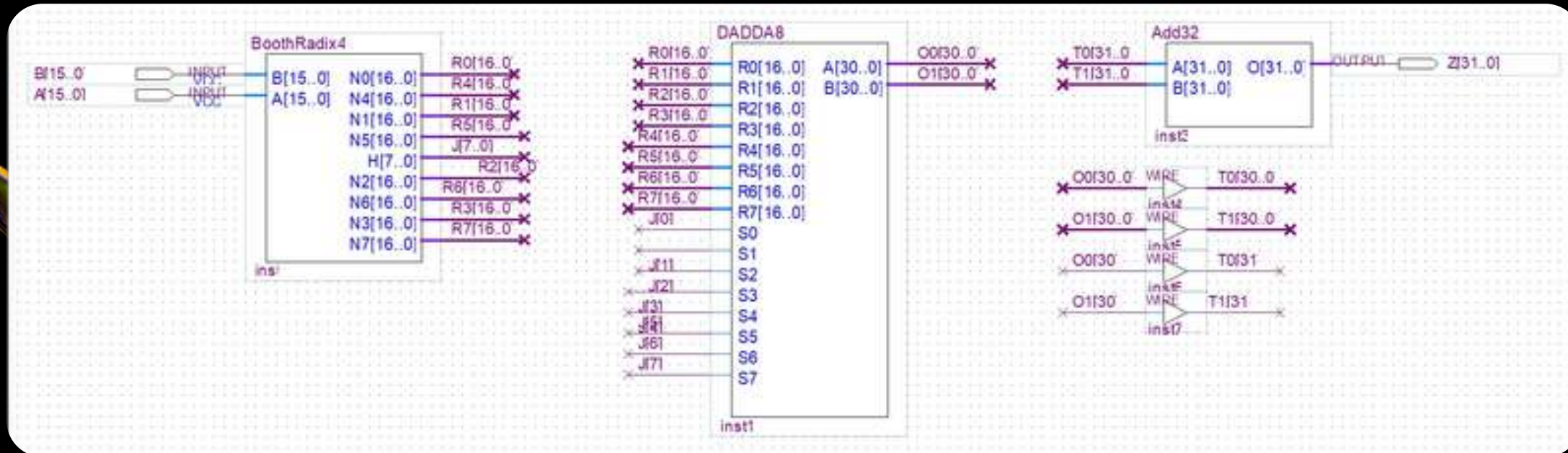
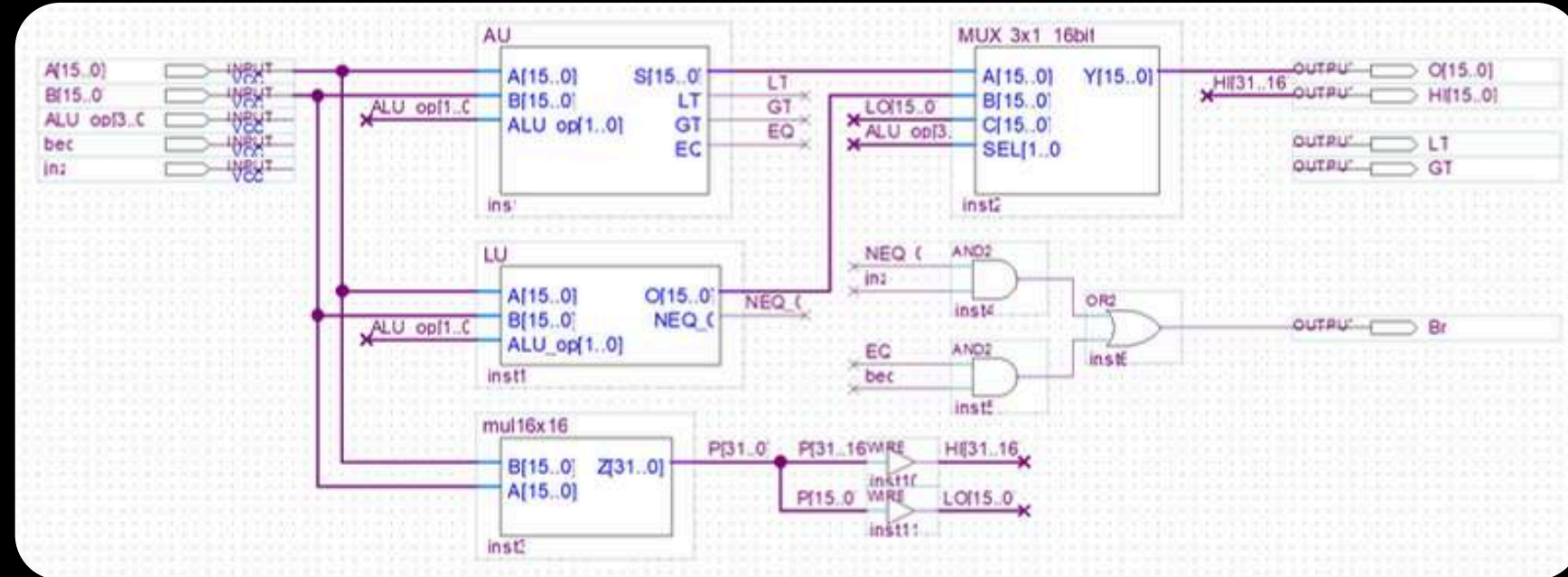
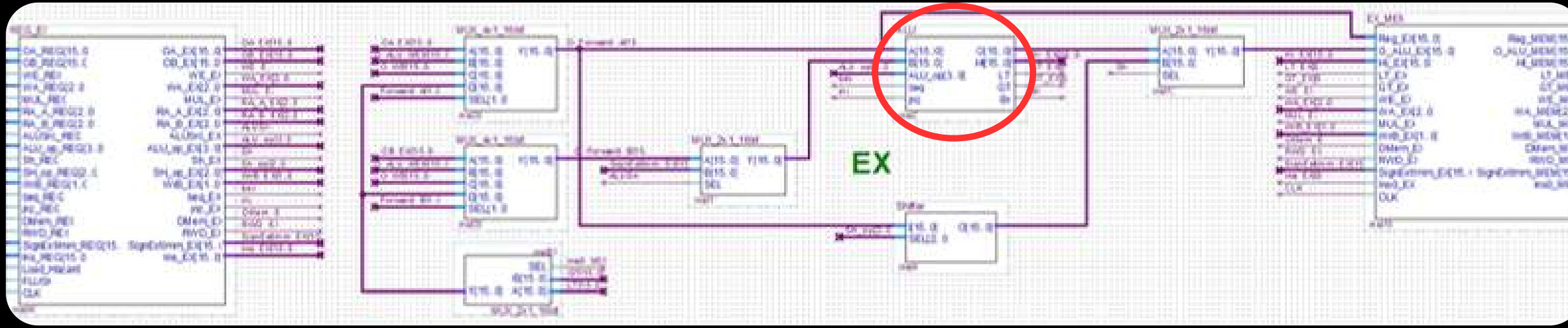
PIPELINE & MẠCH NHẬN



# MẠCH NHÂN

02

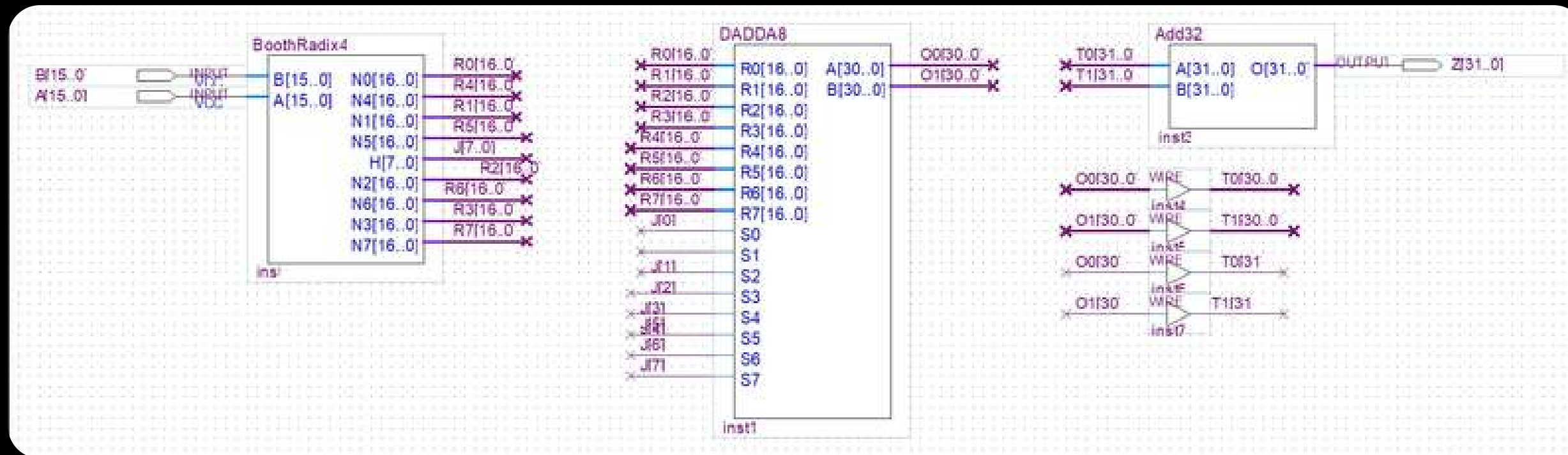
PIPELINE & MẠCH NHÂN



# MẠCH NHÂN

# 02

## PIPELINE & MẠCH NHÂN



→ Thuật toán Booth?

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

**Booth:**  $16 - 1 = 15 \rightarrow 1$  phép toán  
???

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

**Booth (Radix-2)**:  $16 - 1 = 15 \rightarrow 1$  phép toán  
???

0000 1111  
 $i =$  7 6 5 4 3 2 1 0

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

**Booth (Radix-2):**  $16 - 1 = 15 \rightarrow 1$  phép toán  
???

	0	0	0	0	1	1	1	1	0
$i =$	7	6	5	4	3	2	1	0	-1

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

**Booth (Radix-2)**:  $16 \text{ } \boxed{-1} = 15 \rightarrow 1$  phép toán  
???

	0	0	0	0	1	1	1	<b>1</b>	<b>0</b>
$i =$	7	6	5	4	3	2	1	0	-1

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

**Booth (Radix-2)**:  $16 \text{ } \boxed{-1} = 15 \rightarrow 1$  phép toán  
???

	0	0	0	0	1	1	<b>1</b>	<b>1</b>	<b>0</b>
$i =$	7	6	5	4	3	2	1	0	-1

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

**Booth (Radix-2)**:  $16 \text{ } \boxed{-1} = 15 \rightarrow 1$  phép toán  
???

	0	0	0	0	1	1	1	0
$i =$	7	6	5	4	3	2	1	0
								-1

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

**Booth (Radix-2)**:  $16 \text{ } \boxed{-1} = 15 \rightarrow 1$  phép toán  
???

	0	0	0	0	<b>1</b>	<b>1</b>	1	<b>1</b>	<b>0</b>
$i =$	7	6	5	4	3	2	1	0	-1

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

**Booth (Radix-2)**:  $\boxed{16} - \boxed{1} = 15 \rightarrow 1$  phép toán  
???

	0	0	0	0	1	1	1	1	0
$i =$	7	6	5	4	3	2	1	0	-1

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

**Booth (Radix-2)**:  $\boxed{16} - \boxed{1} = 15 \rightarrow 1$  phép toán  
???

000011110  
 $i = 7 \quad 6 \quad 5 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0 \quad -1$

→ Xét từ phải qua trái tại bit  $i$ :

+) 0 → 1:  $-2^i$

+) 1 → 0:  $+2^i$

⇒ Công thức Booth

(**đúng với mọi số có dấu**):

$$\sum_{i=0}^{N-1} (y_{i-1} - y_i) 2^i$$

# MẠCH NHÂN

→ Cách nhân thông thường:

$$\begin{array}{r} 0000\ 0000\ 0000\ 1111\ (15) \\ \times 0111\ 1111\ 1111\ 111\textbf{1}\ (32767) \\ \hline 0000\ \dots\ 0000\ 0000\ 0000\ 1111\ (32\ \text{bits}) \end{array}$$

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

→ Cách nhân thông thường:

$$\begin{array}{r} \phantom{0000} 0000 \ 0000 \ 0000 \ 1111 \ (15) \\ \times \textcircled{0} 111 \ 1111 \ 1111 \ 1111 \ (32767) \\ \hline 0000 \ \dots \ 0000 \ 0000 \ 0000 \ 1111 \\ \phantom{0000} \dots \\ 0000 \ \dots \ 0000 \ 0000 \ 0000 \ 0000 \end{array}$$

} x16

⇒ Phức tạp, không tối ưu.

# MẠCH NHÂN

→ Áp dụng Booth (Booth Radix-2):

$$\begin{array}{r} 0000\ 0000\ 0000\ 1111\ (15) \rightarrow M \\ \times 0111\ 1111\ 1111\ 1111\ 10\ (32767) \\ \hline \end{array}$$

$$-M \ll 0$$

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

→ **Áp dụng Booth (Booth Radix-2):**

Diagram illustrating the bit layout of a 32-bit integer  $x$  for the `rotl` operation:

- The integer  $x$  is represented as a 32-bit word:  $0000\ 0000\ 0000\ 1111\ (15) \rightarrow M$  and  $0111\ 1111\ 1111\ 1111\ 0\ (32767)$ .
- The bit pattern is shown as a sequence of 32 bits, with the first 16 bits being zeros and the last 16 bits being ones.
- The bit pattern is divided into two groups of 16 bits each, labeled  $-M \ll 0$  and  $M \ll 15$ .
- A blue bracket on the right indicates that the bit pattern is repeated 16 times, labeled  $\times 16$ .

⇒ Giảm được độ phức tạp khi cộng nhưng chưa hoàn toàn tối ưu.  
Độ phức tạp không giảm nếu chuỗi bit thay đổi liên tục (TH xấu nhất).

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

**Booth (Radix-4)**:  $16 - 1 = 15 \rightarrow 1$  phép toán

	0	0	0	0	1	1	1	1	0
$i =$	7	6	5	4	3	2	1	0	-1

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

**Booth (Radix-4)**:  $16 - \boxed{1} = 15 \rightarrow 1$  phép toán

	0	0	0	0	1	1	1	0
$i =$	7	6	5	4	3	2	1	0
					-1			

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

**Booth (Radix-4)**:  $\textcircled{16} - \textcircled{1} = 15 \rightarrow 1$  phép toán

$i =$     7   6   5   4   3   2   1   0   -1  
      0 0 0 0 1 1 1 1 0

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

VD: 0000 1111 (15 hệ 10)

Thông thường:  $8 + 4 + 2 + 1 = 15 \rightarrow 3$  phép toán

**Booth (Radix-4)**:  $\boxed{16} - \boxed{1} = 15 \rightarrow 1$  phép toán



**Booth (Radix-2)**:

$$\sum_{i=0}^{N-1} (y_{i-1} - y_i) 2^i$$

$\Rightarrow$

**Booth (Radix-4)**:

$$\sum_{i=0}^{N/2-1} (-2y_{2i+1} + y_{2i} + y_{2i-1}) \cdot 4^i$$

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

$$-2y_{2i+1} + y_{2i} + y_{2i-1}$$

→ Áp dụng Booth (Booth Radix-4):

$$\begin{array}{r} 0000\ 0000\ 0000\ 1111\ (15) \rightarrow M \\ \times 0111\ 1111\ 1111\ 1111\ 0\ (32767) \end{array}$$

---

$$-M \ll 0$$

02

PIPELINE & MẠCH NHÂN

# MẠCH NHÂN

$$-2y_{2i+1} + y_{2i} + y_{2i-1}$$

→ Áp dụng Booth (Booth Radix-4):

$$\begin{array}{r} 0000\ 0000\ 0000\ 1111\ (15) \rightarrow M \\ \times \quad \boxed{0111}\ 1111\ 1111\ 1111\ 0\ (32767) \end{array}$$

$$\begin{array}{r} \hline \phantom{0000} \phantom{...} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \\ \phantom{0000} \phantom{...} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \\ \phantom{0000} \phantom{...} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \\ \phantom{0000} \phantom{...} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \\ \phantom{0000} \phantom{...} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \phantom{0000} \end{array}$$

-M << 0

...

+2M << 14

x8 (Sử dụng Dadda để cộng đồng thời 8 hàng)

⇒ Mức độ tối ưu cao.  
Thuật toán dễ dàng thiết kế.

02

PIPELINE & MẠCH NHÂN



03

---

**KIỂM ĐỊNH**

# KIỂM TRA TÀI NGUYÊN

## Analysis & Synthesis Resource Usage Summary

	Resource	Usage
1	Estimated Total logic elements	3,128
2		
3	Total combinational functions	2793
4	Logic element usage by number of LUT inputs	
1	-- 4 input functions	1851
2	-- 3 input functions	842
3	-- <=2 input functions	100
5		
6	Logic elements by mode	
1	-- normal mode	2793
2	-- arithmetic mode	0
7		
8	Total registers	541
1	-- Dedicated logic registers	541
2	-- I/O registers	0
9		
10	I/O pins	210
11	Embedded Multiplier 9-bit elements	0
12	Maximum fan-out node	CLK
13	Maximum fan-out	541
14	Total fan-out	11779
15	Average fan-out	3.32

Entity	Logic Cells	Dedicated Logic Registers	I/O Registers	Memory Bits	MMIOs	DSP Elements	DSP 9-bit	DSP 18x18	Pins	Virtual Pins	LUT-Only LUTs	Register-Only LUTs	LUT/Register LUTs
Cydonia II: EPF10K10-100	2920 (0)	511 (0)	0 (0)	0	0	0	0	0	210	0	2009 (0)	79 (0)	541 (0)
CPU_10000000	1956 (41)	511 (0)	0 (0)	0	0	0	0	0	0	0	1945 (41)	79 (0)	541 (0)
Control_Unit.m4	1 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	1 (0)	0 (0)	1 (0)
RP_20_00000000	192 (0)	128 (0)	0 (0)	0	0	0	0	0	0	0	49 (49)	0 (0)	143 (130)
MUL_3x1_10000000	32 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	2 (0)	0 (0)	30 (0)
Accum.m4	833 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	751 (0)	0 (0)	84 (0)
Accum.m4	74 (14)	0 (0)	0 (0)	0	0	0	0	0	0	0	74 (14)	0 (0)	0 (0)
LUT.m4	24 (12)	0 (0)	0 (0)	0	0	0	0	0	0	0	23 (13)	0 (0)	1 (1)
MUL_3x1_10000000	755 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	452 (0)	0 (0)	83 (0)
Shift.m4	1 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	1 (0)	0 (0)	0 (0)
MUL_3x1_10000000	79 (0)	11 (0)	0 (0)	0	0	0	0	0	0	0	62 (0)	0 (0)	17 (0)
PC.m4	29 (18)	0 (0)	0 (0)	0	0	0	0	0	0	0	18 (0)	0 (0)	11 (0)
SAD.m4	79 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	70 (0)	0 (0)	9 (0)
MUL_3x1_10000000	74 (12)	53 (14)	0 (0)	0	0	0	0	0	0	0	16 (16)	0 (0)	38 (38)
ID.m4	362 (0)	145 (0)	0 (0)	0	0	0	0	0	0	0	221 (0)	12 (0)	109 (1)
RP_00000000	138 (13)	128 (0)	0 (0)	0	0	0	0	0	0	0	216 (13)	38 (0)	48 (0)
Update.m4	4 (4)	0 (0)	0 (0)	0	0	0	0	0	0	0	2 (2)	0 (0)	2 (2)
CLA.m4	4 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	1 (0)	0 (0)	1 (0)
PC_Storage.m4	12 (0)	12 (0)	0 (0)	0	0	0	0	0	0	0	0 (0)	2 (0)	2 (0)
MUL_3x1_10000000	8 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	4 (0)	0 (0)	0 (0)
EX.m4	146 (4)	132 (4)	0 (0)	0	0	0	0	0	0	0	18 (0)	4 (0)	138 (4)
MEM.m4	46 (2)	46 (2)	0 (0)	0	0	0	0	0	0	0	0 (0)	13 (1)	53 (3)
MUL_3x1_10000000	46 (0)	11 (0)	0 (0)	0	0	0	0	0	0	0	35 (0)	0 (0)	35 (0)
Forwarding_Unit.m4	25 (25)	18 (14)	0 (0)	0	0	0	0	0	0	0	11 (13)	0 (0)	14 (14)
Load_Register_Definition_Unit.m4	1 (0)	0 (0)	0 (0)	0	0	0	0	0	0	0	0 (0)	0 (0)	1 (1)
MUL_3x1_10000000	18 (0)	1 (0)	0 (0)	0	0	0	0	0	0	0	17 (0)	0 (0)	1 (0)
MUL_3x1_10000000	105 (0)	42 (0)	0 (0)	0	0	0	0	0	0	0	23 (23)	16 (2)	86 (46)
MUL_3x1_10000000	1 (0)	1 (0)	0 (0)	0	0	0	0	0	0	0	0 (0)	0 (0)	1 (1)
OutCtrl.m4	29 (29)	0 (0)	0 (0)	0	0	0	0	0	0	0	14 (14)	0 (0)	13 (13)
ID.m4	634 (2)	0 (0)	0 (0)	0	0	0	0	0	0	0	932 (2)	0 (0)	2 (0)
ID_Controller.m4	18 (2)	0 (0)	0 (0)	0	0	0	0	0	0	0	18 (2)	0 (0)	0 (0)

# KIỂM TRA ĐỊNH THỜI

Slow Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	60.14 MHz	60.14 MHz	clk	

Slow Model Setup Summary

	Clock	Slack	End Point TNS
1	clk	3.373	0.000

Slow Model Hold Summary

	Clock	Slack	End Point TNS
1	clk	0.516	0.000

# KIỂM TRA ĐỊNH THỜI

## Slow Model Setup Summary

	Clock	Slack	End Point TNS
1	clk	3.373	0.000

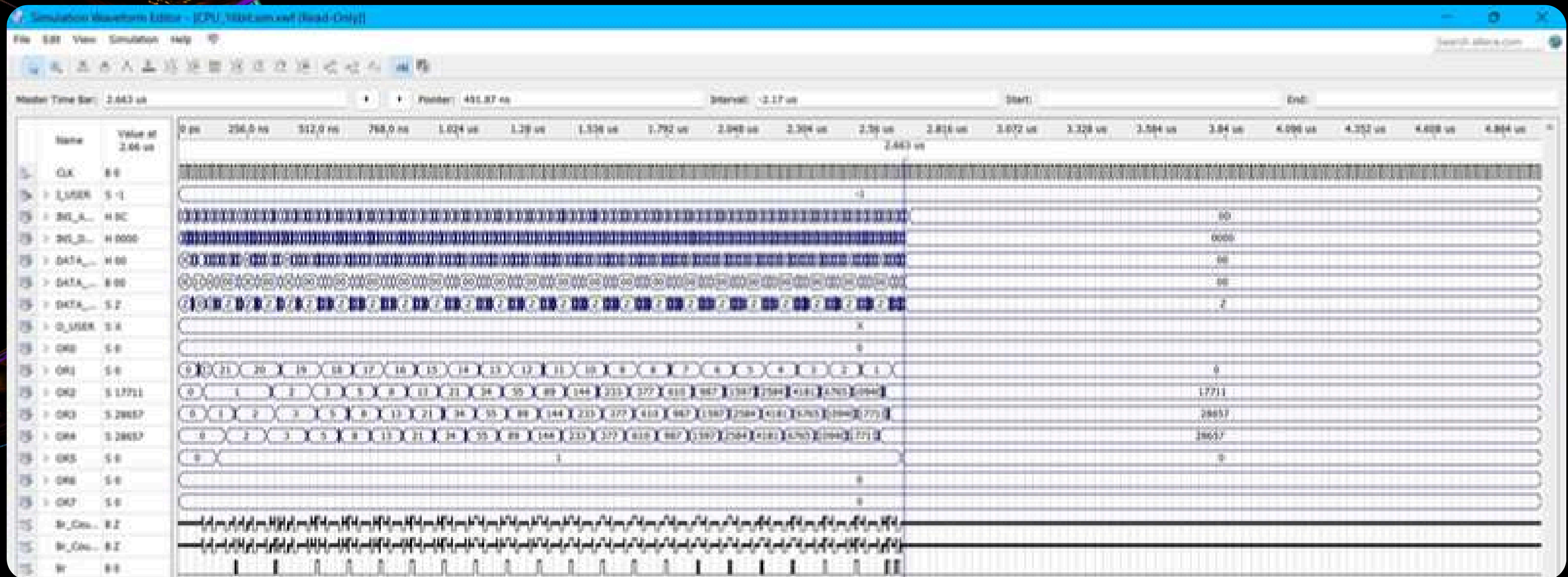
## Slow Model Hold Summary

	Clock	Slack	End Point TNS
1	clk	0.516	0.000

Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
3.373	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst11 inst15	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...bit:inst2 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.001	16.662
3.373	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...bit:inst2 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.001	16.662
3.615	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst11 inst15	CPU_16bit:inst Forwarding_Unit:inst20 inst1~3_OTERM188	clk	clk	20.000	-0.003	16.418
3.615	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst Forwarding_Unit:inst20 inst1~3_OTERM188	clk	clk	20.000	-0.003	16.418
3.641	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst11 inst15	CPU_16bit:inst MUX_2x1_16bit:inst11 MUX...bit:inst2 MUX_2x1:inst2 inst~0_OTERM674	clk	clk	20.000	-0.002	16.393
3.641	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_2x1_16bit:inst11 MUX...bit:inst2 MUX_2x1:inst2 inst~0_OTERM674	clk	clk	20.000	-0.002	16.393
3.656	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...bit:inst2 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.001	16.379
3.755	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst24 inst13	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...bit:inst2 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	0.008	16.289
3.834	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst11 inst15	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...bit:inst2 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.001	16.201
3.898	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst Forwarding_Unit:inst20 inst1~3_OTERM188	clk	clk	20.000	-0.003	16.135
3.924	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_2x1_16bit:inst11 MUX...bit:inst2 MUX_2x1:inst2 inst~0_OTERM674	clk	clk	20.000	-0.002	16.110
3.941	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst11 inst15	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...it:inst3 MUX_2x1:inst1 inst3~1_OTERM136	clk	clk	20.000	-0.003	16.092
3.941	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...it:inst3 MUX_2x1:inst1 inst3~1_OTERM136	clk	clk	20.000	-0.003	16.092
3.945	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst11 inst15	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...bit:inst2 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	0.003	16.094
3.997	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst24 inst13	CPU_16bit:inst Forwarding_Unit:inst20 inst1~3_OTERM188	clk	clk	20.000	0.006	16.045
4.023	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst24 inst13	CPU_16bit:inst MUX_2x1_16bit:inst11 MUX...bit:inst2 MUX_2x1:inst2 inst~0_OTERM674	clk	clk	20.000	0.007	16.020
4.044	CPU_16bit:inst EX_MEM:inst16 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...bit:inst2 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	0.000	15.992
4.076	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst11 inst15	CPU_16bit:inst Forwarding_Unit:inst20 inst1~3_OTERM188	clk	clk	20.000	-0.003	15.957
4.102	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst11 inst15	CPU_16bit:inst MUX_2x1_16bit:inst11 MUX...bit:inst2 MUX_2x1:inst2 inst~0_OTERM674	clk	clk	20.000	-0.002	15.932
4.102	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst24 inst13	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...bit:inst2 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.115	15.819
4.104	CPU_16bit:inst RF_20_8x16:inst1 RFC_20..._at1 RFC_20_4bit:inst1 RFC_20:inst inst4	CPU_16bit:inst MUX_5x1_16bit:inst18 M...UX_2x1:inst1 inst3~1_OTERM126_OTERM594	clk	clk	10.000	0.008	5.940
4.110	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst2 inst15	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...bit:inst2 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.001	15.925
4.123	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst11 inst15	CPU_16bit:inst MUX_2x1_16bit:inst11 MUX...bit:inst2 MUX_2x1:inst2 inst~1_OTERM664	clk	clk	20.000	0.002	15.915
4.123	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_2x1_16bit:inst11 MUX...bit:inst2 MUX_2x1:inst2 inst~1_OTERM664	clk	clk	20.000	0.002	15.915
4.124	CPU_16bit:inst Forwarding_Unit:inst20 inst22~2_OTERM124	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...bit:inst2 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.003	15.909
4.131	CPU_16bit:inst RF_20_8x16:inst1 RFC_20..._at1 RFC_20_4bit:inst1 RFC_20:inst3 inst4	CPU_16bit:inst MUX_5x1_16bit:inst18 M...X_2x1:inst2 inst3~3_OTERM502_OTERM526	clk	clk	10.000	0.002	5.907
4.166	CPU_16bit:inst Forwarding_Unit:inst20 inst~1_OTERM122	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...bit:inst2 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	0.002	15.872
4.187	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst11 inst14	CPU_16bit:inst Forwarding_Unit:inst20 inst1~3_OTERM188	clk	clk	20.000	0.001	15.850
4.203	CPU_16bit:inst MUX_2x1_16bit:inst11 M...UX_2x1:inst1 inst~2_OTERM196_OTERM514	CPU_16bit:inst MUX_5x1_16bit:inst18 MUX...bit:inst2 MUX_2x1:inst1 inst3~0_OTERM660	clk	clk	20.000	-0.002	15.831
4.204	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst11 inst15	CPU_16bit:inst MUX_2x1_16bit:inst11 M...UX_4bit:inst3 MUX_2x1:inst1 inst~0_OTERM666	clk	clk	20.000	0.001	15.833
4.204	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst MUX_2x1_16bit:inst11 M...UX_4bit:inst3 MUX_2x1:inst1 inst~0_OTERM666	clk	clk	20.000	0.001	15.833
4.205	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst11 inst15	CPU_16bit:inst Forwarding_Unit:inst20 inst2~3_OTERM202	clk	clk	20.000	0.001	15.832
4.205	CPU_16bit:inst MEM_WB:inst17 DFF_2bit:inst11 inst1	CPU_16bit:inst Forwarding_Unit:inst20 inst2~3_OTERM202	clk	clk	20.000	0.001	15.832

Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
0.516	CPU_16bit:inst ID_REG:inst14 DFF_3bit:inst2 inst2	CPU_16bit:inst REG_EX:inst40 DFF_3bit:inst4 inst2_OTERM432	clk	clk	0.000	0.000	0.782
0.516	CPU_16bit:inst ID_REG:inst14 inst18	CPU_16bit:inst REG_EX:inst40 inst15_OTERM462	clk	clk	0.000	0.000	0.782
0.516	CPU_16bit:inst ID_REG:inst14 DFF_16bit:inst1 inst12	CPU_16bit:inst REG_EX:inst40 DFF_16bit:inst3 inst12	clk	clk	0.000	0.000	0.782
0.517	CPU_16bit:inst REG_EX:inst40 DFF_3bit:inst4 inst_OTERM444	CPU_16bit:inst EX_MEM:inst16 DFF_3bit:inst4 inst	clk	clk	0.000	0.000	0.783
0.518	CPU_16bit:inst ID_REG:inst14 inst4	CPU_16bit:inst REG_EX:inst40 inst6_OTERM440	clk	clk	0.000	0.000	0.784
0.518	CPU_16bit:inst REG_EX:inst40 DFF_3bit:inst4 inst1_OTERM442	CPU_16bit:inst EX_MEM:inst16 DFF_3bit:inst4 inst1	clk	clk	0.000	0.000	0.784
0.518	CPU_16bit:inst ID_REG:inst14 inst17_OTERM420	CPU_16bit:inst REG_EX:inst40 inst14_OTERM460	clk	clk	0.000	0.000	0.784
0.519	CPU_16bit:inst ID_REG:inst14 inst5_OTERM412	CPU_16bit:inst REG_EX:inst40 inst7_OTERM446	clk	clk	0.000	0.000	0.785
0.519	CPU_16bit:inst ID_REG:inst14 DFF_16bit:inst19 inst4	CPU_16bit:inst REG_EX:inst40 DFF_16bit:inst18 inst4	clk	clk	0.000	0.000	0.785
0.520	CPU_16bit:inst ID_REG:inst14 DFF_16bit:inst1 inst10	CPU_16bit:inst REG_EX:inst40 DFF_16bit:inst3 inst10	clk	clk	0.000	0.000	0.786
0.520	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst25 inst15_OTERM392	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst25 inst15	clk	clk	0.000	0.000	0.786
0.523	CPU_16bit:inst ID_REG:inst14 DFF_16bit:inst19 inst5	CPU_16bit:inst REG_EX:inst40 DFF_16bit:inst18 inst5	clk	clk	0.000	0.000	0.789
0.527	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst inst	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst inst	clk	clk	0.000	0.000	0.793
0.530	CPU_16bit:inst ID_REG:inst14 DFF_16bit:inst19 inst6	CPU_16bit:inst REG_EX:inst40 DFF_16bit:inst18 inst6	clk	clk	0.000	0.000	0.796
0.531	CPU_16bit:inst BranchPrediction:inst15 PC...age:inst9 RF_1x6_10:inst3 RFC:inst5 inst6	CPU_16bit:inst BranchPrediction:inst15 PC...age:inst9 RF_1x6_10:inst2 RFC:inst5 inst6	clk	clk	0.000	0.000	0.797
0.539	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst25 inst6_OTERM86	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst25 inst6	clk	clk	0.000	0.000	0.805
0.545	CPU_16bit:inst ID_REG:inst14 DFF_3bit:inst6 inst2	CPU_16bit:inst REG_EX:inst40 DFF_3bit:inst6 inst2	clk	clk	0.000	0.000	0.811
0.548	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst25 inst10_OTERM236	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst25 inst10	clk	clk	0.000	0.000	0.815
0.550	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst25 inst1_OTERM3	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst25 inst1	clk	clk	0.000	0.000	0.816
0.550	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst25 inst1_OTERM3	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst25 inst	clk	clk	0.000	0.000	0.816
0.555	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst25 inst8_OTERM178	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst25 inst8	clk	clk	0.000	0.000	0.821
0.655	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst25 inst9_OTERM230	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst25 inst9	clk	clk	0.000	0.000	0.921
0.657	CPU_16bit:inst ID_REG:inst14 DFF_2bit:inst11 inst	CPU_16bit:inst REG_EX:inst40 DFF_2bit:inst11 inst_OTERM436	clk	clk	0.000	0.000	0.923
0.658	CPU_16bit:inst ID_REG:inst14 DFF_16bit:inst25 inst15	CPU_16bit:inst REG_EX:inst40 DFF_16bit:inst18 inst15	clk	clk	0.000	0.000	0.924
0.660	CPU_16bit:inst ID_REG:inst14 DFF_16bit:inst19 inst11	CPU_16bit:inst REG_EX:inst40 DFF_16bit:inst18 inst11	clk	clk	0.000	0.000	0.926
0.662	CPU_16bit:inst BranchPrediction:inst15 PC...age:inst9 RF_1x6_10:inst3 RFC:inst6 inst6	CPU_16bit:inst BranchPrediction:inst15 PC...age:inst9 RF_1x6_10:inst2 RFC:inst6 inst6	clk	clk	0.000	0.000	0.928
0.662	CPU_16bit:inst BranchPrediction:inst15 PC...age:inst9 RF_1x6_10:inst3 RFC:inst inst6	CPU_16bit:inst BranchPrediction:inst15 PC...age:inst9 RF_1x6_10:inst2 RFC:inst inst6	clk	clk	0.000	0.000	0.928
0.662	CPU_16bit:inst ID_REG:inst14 DFF_16bit:inst19 inst2	CPU_16bit:inst REG_EX:inst40 DFF_16bit:inst18 inst2	clk	clk	0.000	0.000	0.928
0.663	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst24 inst14	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst inst14	clk	clk	0.000	0.000	0.929
0.665	CPU_16bit:inst BranchPrediction:inst15 PC...age:inst9 RF_1x6_10:inst3 RFC:inst3 inst6	CPU_16bit:inst BranchPrediction:inst15 PC...age:inst9 RF_1x6_10:inst2 RFC:inst3 inst6	clk	clk	0.000	0.000	0.931
0.665	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst25 inst2_OTERM440	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst25 inst2	clk	clk	0.000	0.000	0.931
0.665	CPU_16bit:inst EX_MEM:inst16 DFF_16bit:inst25 inst3_OTERM288	CPU_16bit:inst MEM_WB:inst17 DFF_16bit:inst25 inst3	clk	clk	0.000	0.000	0.931
0.666	CPU_16bit:inst BranchPrediction:inst15 PC...age:inst9 RF_1x6_10:inst3 RFC:inst4 inst6	CPU_16bit:inst BranchPrediction:inst15 PC...age:inst9 RF_1x6_10:inst2 RFC:inst4 inst6	clk	clk	0.000	0.000	0.932

# KIỂM TRA ĐỊNH THỜI



# KIỂM TRA CÔNG SUẤT

## PowerPlay Power Analyzer Summary

PowerPlay Power Analyzer Status	Successful - Wed Dec 17 18:02:14 2025
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	CPU_16bit
Top-level Entity Name	Datapath
Family	Cyclone II
Device	EP2C35F672C6
Power Models	Final
Total Thermal Power Dissipation	217.47 mW
Core Dynamic Thermal Power Dissipation	23.72 mW
Core Static Thermal Power Dissipation	80.28 mW
I/O Thermal Power Dissipation	113.47 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

# TỔNG HỢP KẾT QUẢ

Tiêu chí	CPU 16-bit	Nios II/s (Standard)
Tài nguyên	- Logic elements: <b>3,128</b> - Registers: <b>541</b>	LEs: $\approx$ <b>1,400 - 1,800</b>
Tần số tối đa	- $F_{\max} \approx$ <b>60.14 MHz</b> - $T_{\min} \approx$ <b>16.63 ns</b>	$\approx$ <b>80 - 100 MHz</b> (trên Cyclone II)
Định thời	- Setup slack: <b>3.373 ns</b> - Hold slack: <b>0.516 ns</b>	...
Công suất	- Total Thermal Power Dissipation: <b>217.47 mW</b> - Core Dynamic Thermal Power Dissipation: <b>23.72 mW</b> - Core Static Thermal Power Dissipation: <b>80.28 mW</b> - I/O Thermal Power Dissipation: <b>113.47 mW</b>	- Core Dynamic Thermal Power Dissipation: <b>10 - 15 mW</b>
Hiệu năng (Testbench: Tìm số Fibonacci thứ 23)	- Thời gian thực thi: $\approx$ <b>2.663 <math>\mu</math>s</b> - Testbench: li r1, 23 li r2, 1 li r3, 1 subi r1, r1, 2 slt r5, r0, r1 beq r5, r0, 11 add r4, r3, r2 mov r2, r3 mov r3, r4 dec r1, r1 jmp 4	- Thời gian thực thi: $\approx$ <b>2.5 - 3.5 <math>\mu</math>s</b>

---

Nhóm 1 - CE118.Q11.VMTN

# THANK YOU

for your time and attention

**GitHub Project:**

