

People divided into 10 types –  
those who understand binary code  
and those who do not



# Архитектуры процессорных систем

## Лекция 3. Цифровая арифметика. АЛУ

Цикл из 16 лекций о цифровой схемотехнике, способах построения и архитектуре компьютеров

# План лекции

- Двоичное и шестнадцатеричное представление чисел
- Двоичная арифметика
- Отрицательные числа
- Сумматор
- Арифметико-логическое устройство

# Двоичное представление чисел

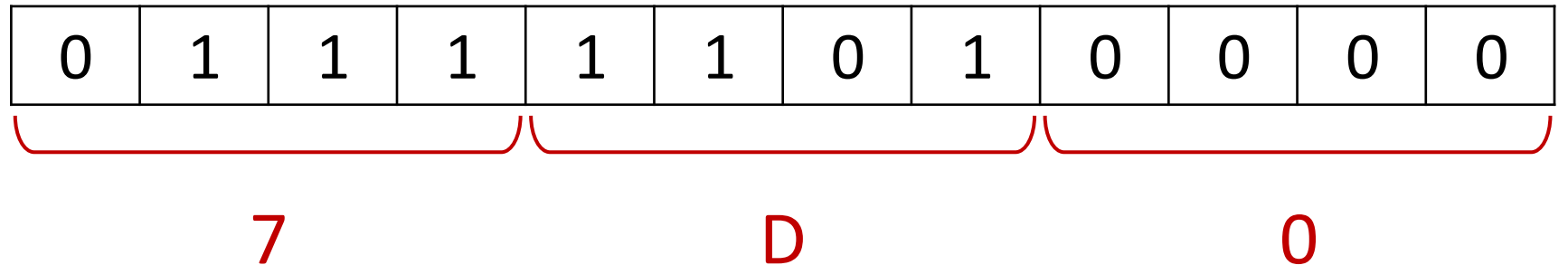
$$v = \sum_{i=0}^{N-1} 2^i b_i$$

512	256	128	64	32	16	8	4	2	1
$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
0	1	1	0	1	1	1	1	0	0

$$\begin{aligned} v &= 0 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + \dots \\ &= 256 + 128 + 32 + 16 + 8 + 4 = 444 \end{aligned}$$

# Шестнадцатеричное представление чисел

0000 – 0  
0001 – 1  
0010 – 2  
0011 – 3  
0100 – 4  
0101 – 5  
0110 – 6  
0111 – 7  
1000 – 8  
1001 – 9  
1010 – A  
1011 – B  
1100 – C  
1101 – D  
1110 – E  
1111 – F



**0b01111010000 = 0x7D0**

# Двоичная арифметика

$$\begin{array}{r} \text{1} \text{ --- carry} \\ + 14 \\ + 7 \\ \hline 21 \end{array}$$

$$\begin{array}{r} 111 \\ + 1110 \\ + 111 \\ \hline 10101 \end{array}$$

$$-2^3 + 0b110 = -8 + 6 = -2$$

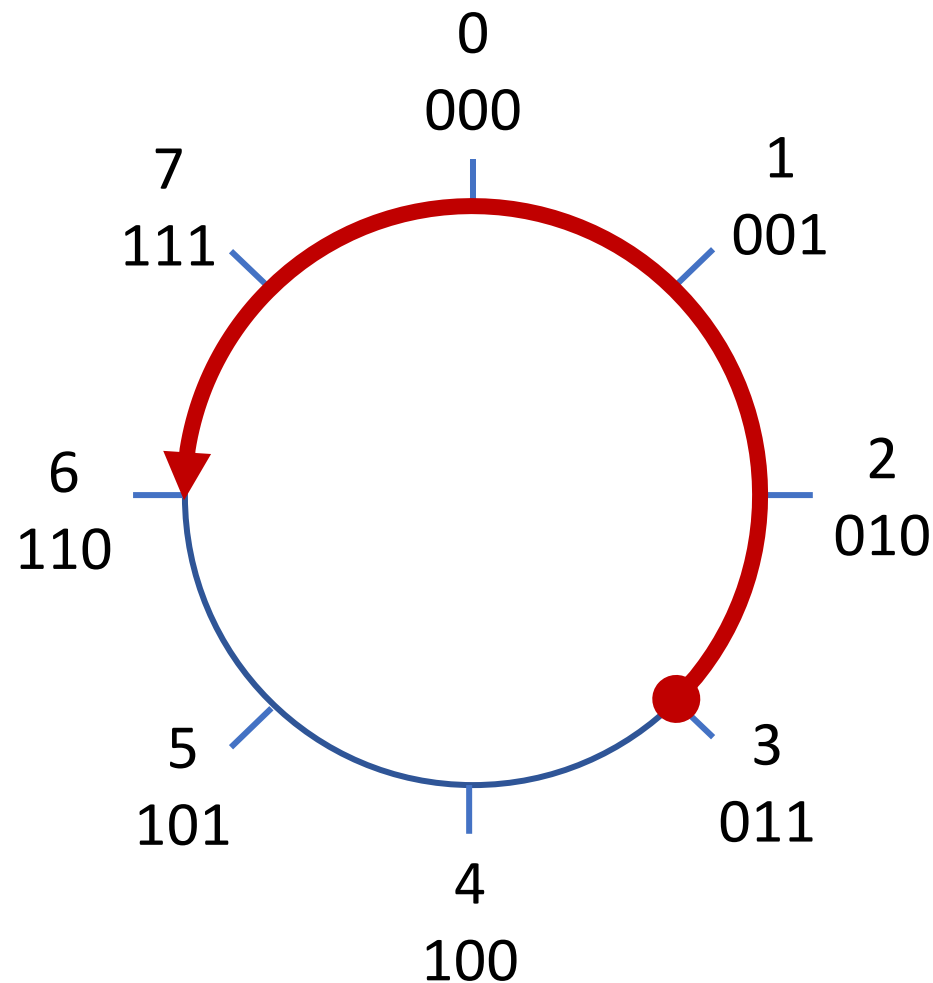
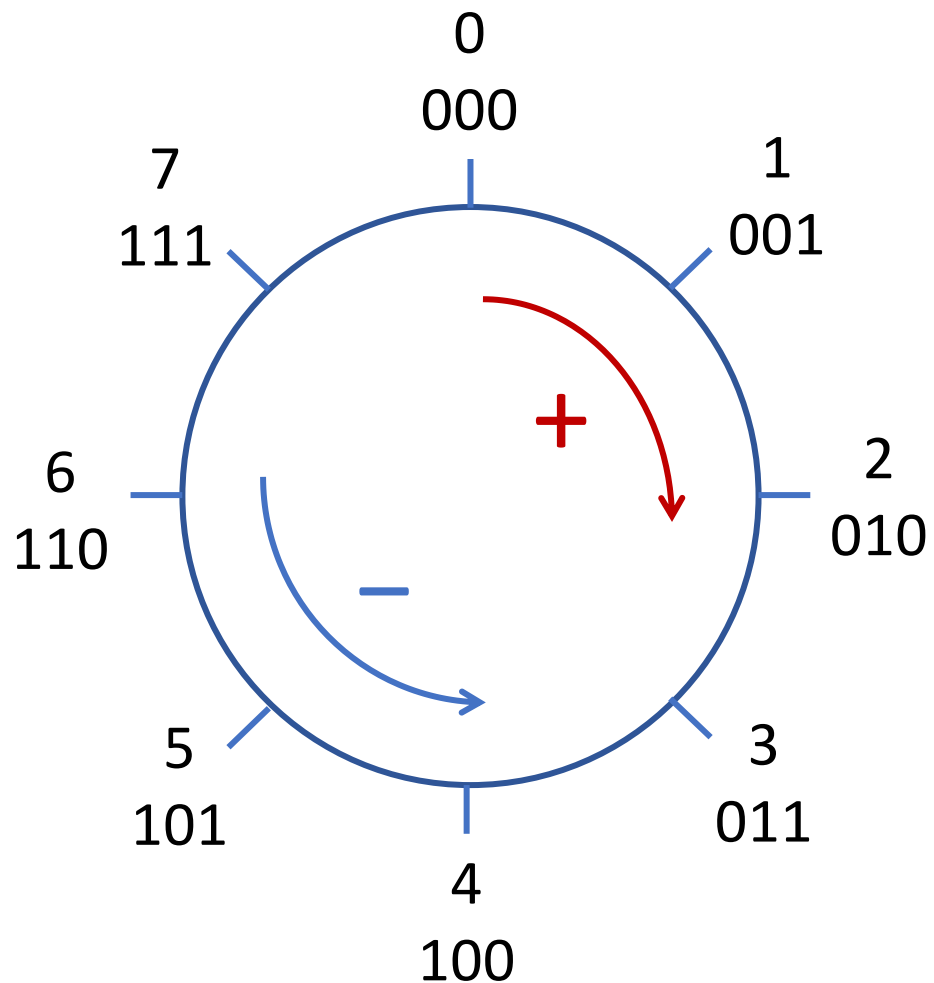
$$\begin{array}{r} -1 \text{ --- borrow} \\ - 14 \\ - 7 \\ \hline 07 \end{array}$$

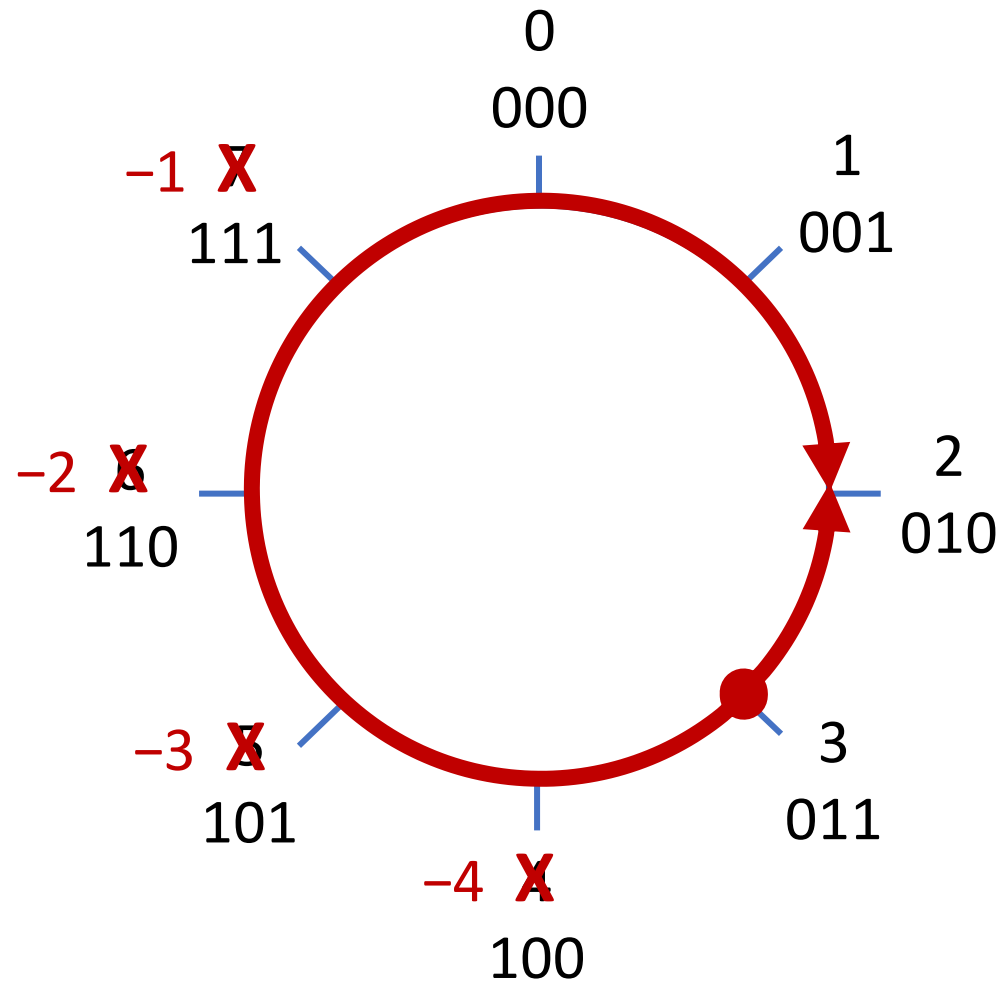
$$\begin{array}{r} -1-1-1 \\ - 1110 \\ - 111 \\ \hline 0111 \end{array}$$

$$\begin{array}{r} -1 \\ - 011 \\ - 101 \\ \hline ???110 \end{array}$$

# Арифметика по модулю

$$(3 - 5) \bmod 8 = ?$$







# Числа в дополнительном коде

N бит

$$v = -2^{N-1}b_{N-1} + \sum_{i=0}^{N-2} 2^i b_i$$



- Отрицательные числа имеют «1» в старшем разряде
- Самое маленькое число?      **1**0...0000       $-2^{N-1}$
- Самое большое число?      **0**1...1111       $+2^{N-1} - 1$
- Если все биты «1»?      11...1111       $-1$

# Отрицательные числа

Как положительное число сделать отрицательным?

Как из  $A$  сделать  $-A$ ?

$$-A + A = 0 = -1 + 1$$

$$-A = \underbrace{(-1 - A)} + 1$$

$$\begin{array}{r} \phantom{-} \phantom{A_{N-1}} \phantom{\dots} \phantom{A_1} \phantom{A_0} \\ \phantom{-} \phantom{A_{N-1}} \phantom{\dots} \phantom{A_1} \phantom{A_0} \\ \phantom{-} \phantom{A_{N-1}} \phantom{\dots} \phantom{A_1} \phantom{A_0} \\ \phantom{-} \phantom{A_{N-1}} \phantom{\dots} \phantom{A_1} \phantom{A_0} \\ \hline \phantom{-} \phantom{A_{N-1}} \phantom{\dots} \phantom{A_1} \phantom{A_0} \\ \hline \phantom{-} \phantom{A_{N-1}} \phantom{\dots} \phantom{A_1} \phantom{A_0} \end{array}$$

$$-A = \bar{A} + 1$$

$$A - B = A + \bar{B} + 1$$

# Переполнение

$$\begin{array}{r} (-5) \quad 1011 \\ (+5) \quad + 0101 \\ \hline (0) \quad 10000 \end{array}$$

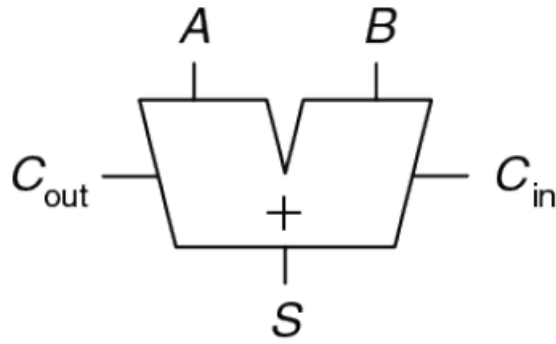
$$\begin{array}{r} (+5) \quad 0101 \\ (+4) \quad + 0100 \\ \hline (-7) \quad 1001 \end{array}$$

$$\begin{array}{r} (-6) \quad 1010 \\ (-5) \quad + 1011 \\ \hline (+5) \quad 10101 \end{array}$$

# Сумматор

$A_3$	$A_2$	$A_1$	$A_0$
$B_3$	$B_2$	$B_1$	$B_0$
<hr/>			
$S_3$	$S_2$	$S_1$	$S_0$

**Full  
Adder**



$C_{in}$	$A$	$B$	$C_{out}$	$S$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

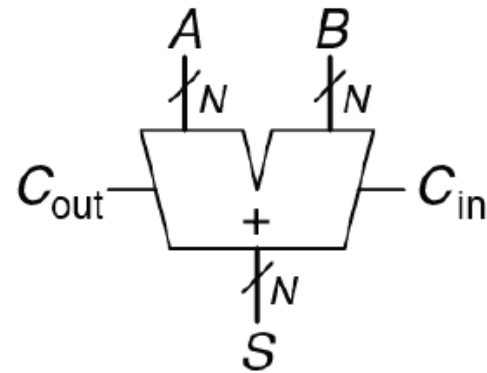
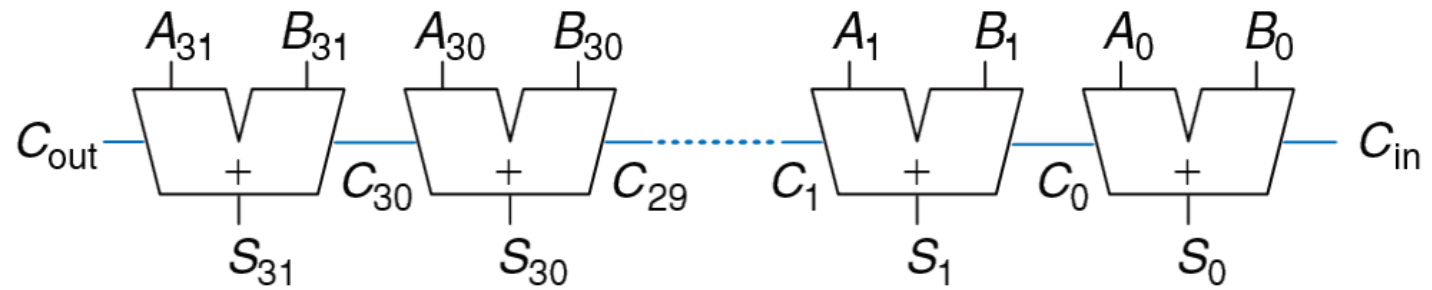
$$S = A \oplus B \oplus C_{in}$$

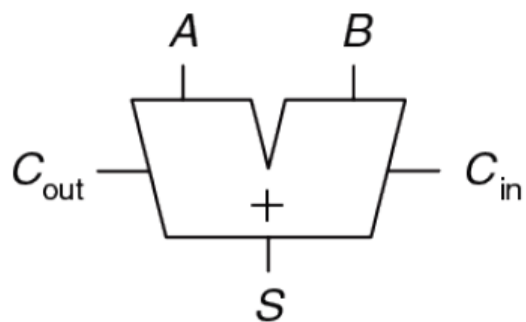
$$C_{out} = AB + AC_{in} + BC_{in}$$

$x$	$y$	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

# Сумматор

$$\begin{array}{r} A_3 A_2 A_1 A_0 \\ B_3 B_2 B_1 B_0 \\ \hline S_3 S_2 S_1 S_0 \end{array}$$





A	B	C <sub>in</sub>	C <sub>out</sub>	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$C_i = A_i B_i + (A_i + B_i) C_{i-1}$$

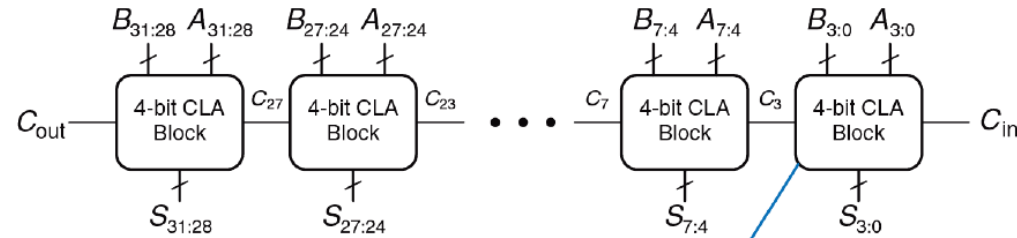
$$C_i = A_i B_i + (A_i + B_i) C_{i-1} = G_i + P_i C_{i-1}$$

$$G_{3:0} = G_3 + P_3 (G_2 + P_2 (G_1 + P_1 G_0))$$

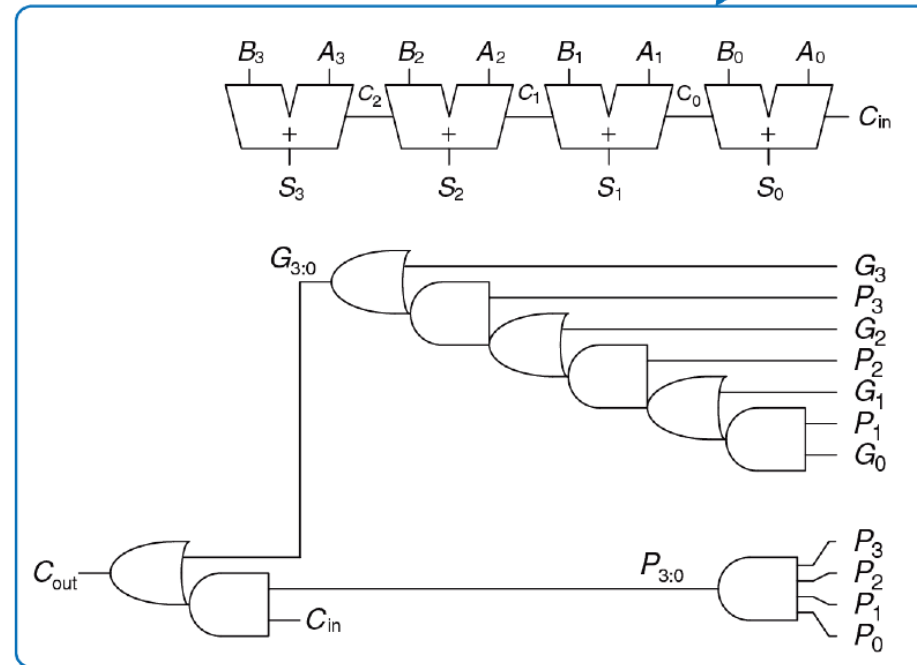
$$P_{3:0} = P_3 P_2 P_1 P_0$$

$$C_i = G_{i:j} + P_{i:j} C_j$$

# Сумматор с ускоренным переносом



(a)



(b)

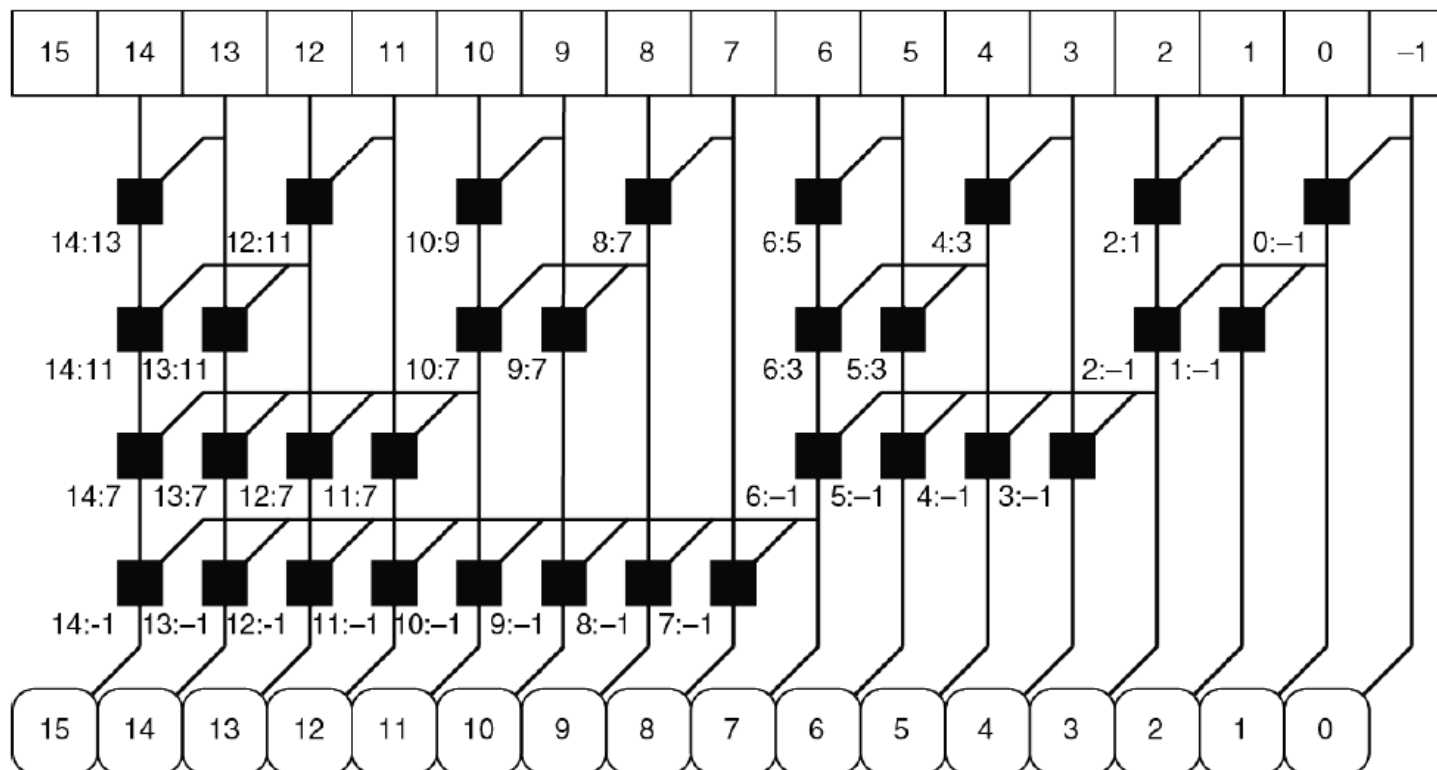
$$C_i = G_{i,j} + P_{i,j} C_j$$

$$G_{3:0} = G_3 + P_3 (G_2 + P_2 (G_1 + P_1 G_0))$$

$$P_{3:0} = P_3 P_2 P_1 P_0$$



# Префиксный сумматор

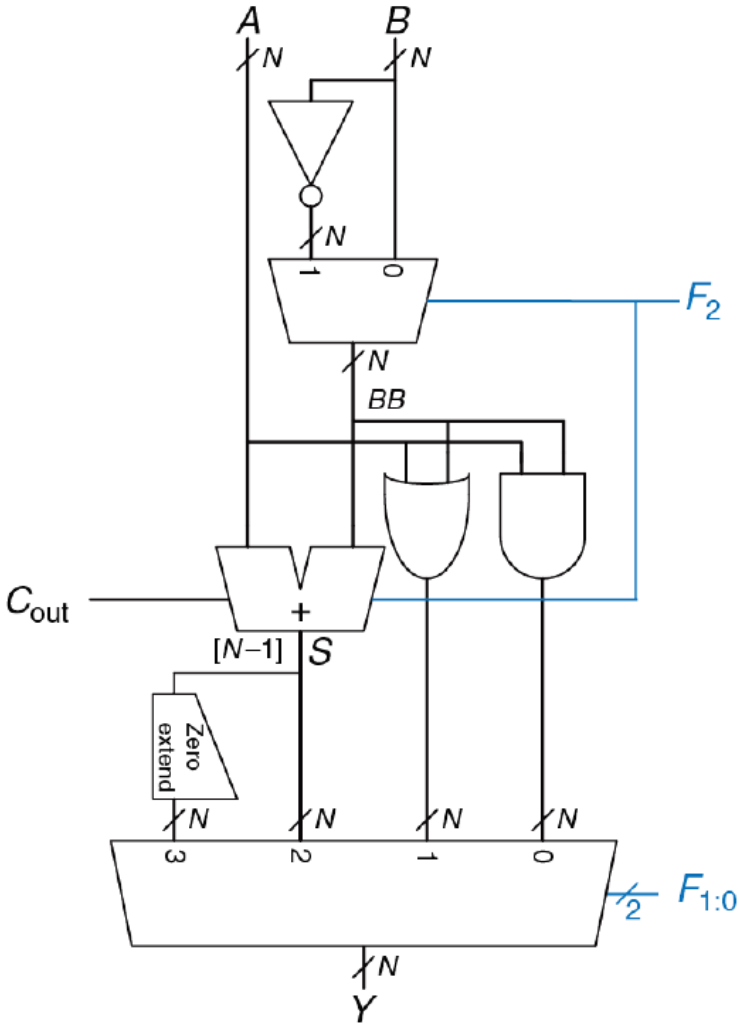


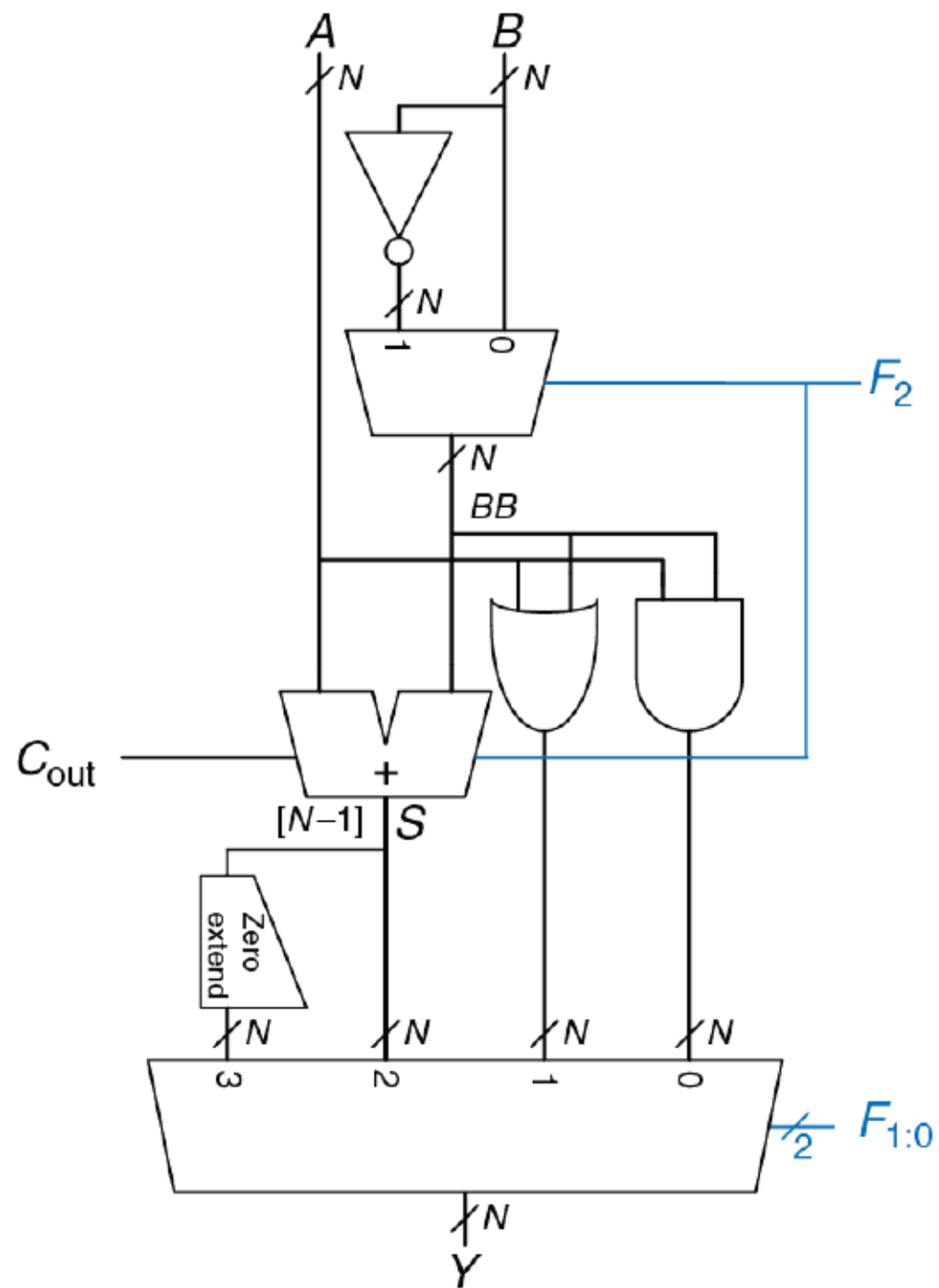
ALU

# Арифметико-логическое устройство

- АЛУ – блок процессора, выполняющий арифметические и поразрядно логические операции
  - Арифметические операции имеют перенос
  - Логические операции без переноса
- АЛУ – комбинационная схема
- На вход АЛУ поступают **информационные** сигналы (данные, над которыми происходит операция) и **управляющие** сигналы (определяют, какая операция будет произведена над данными), на выходе – **результат** операции и **флаги**

# АЛУ MIPS





$F_{2:0}$	Function
000	A AND B
001	A OR B
010	A + B
011	not used
100	A AND $\bar{B}$
101	A OR $\bar{B}$
110	A - B
111	SLT

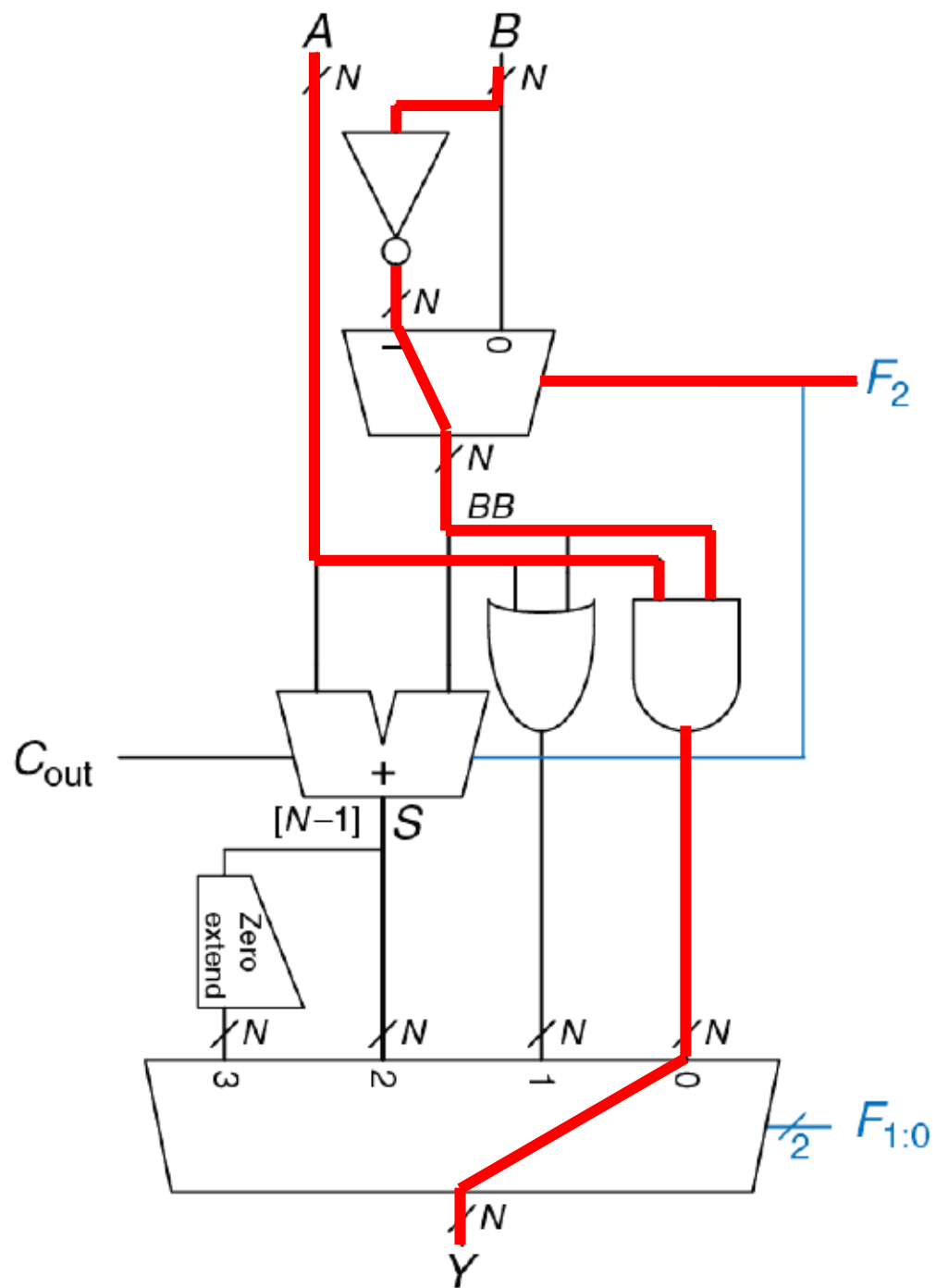




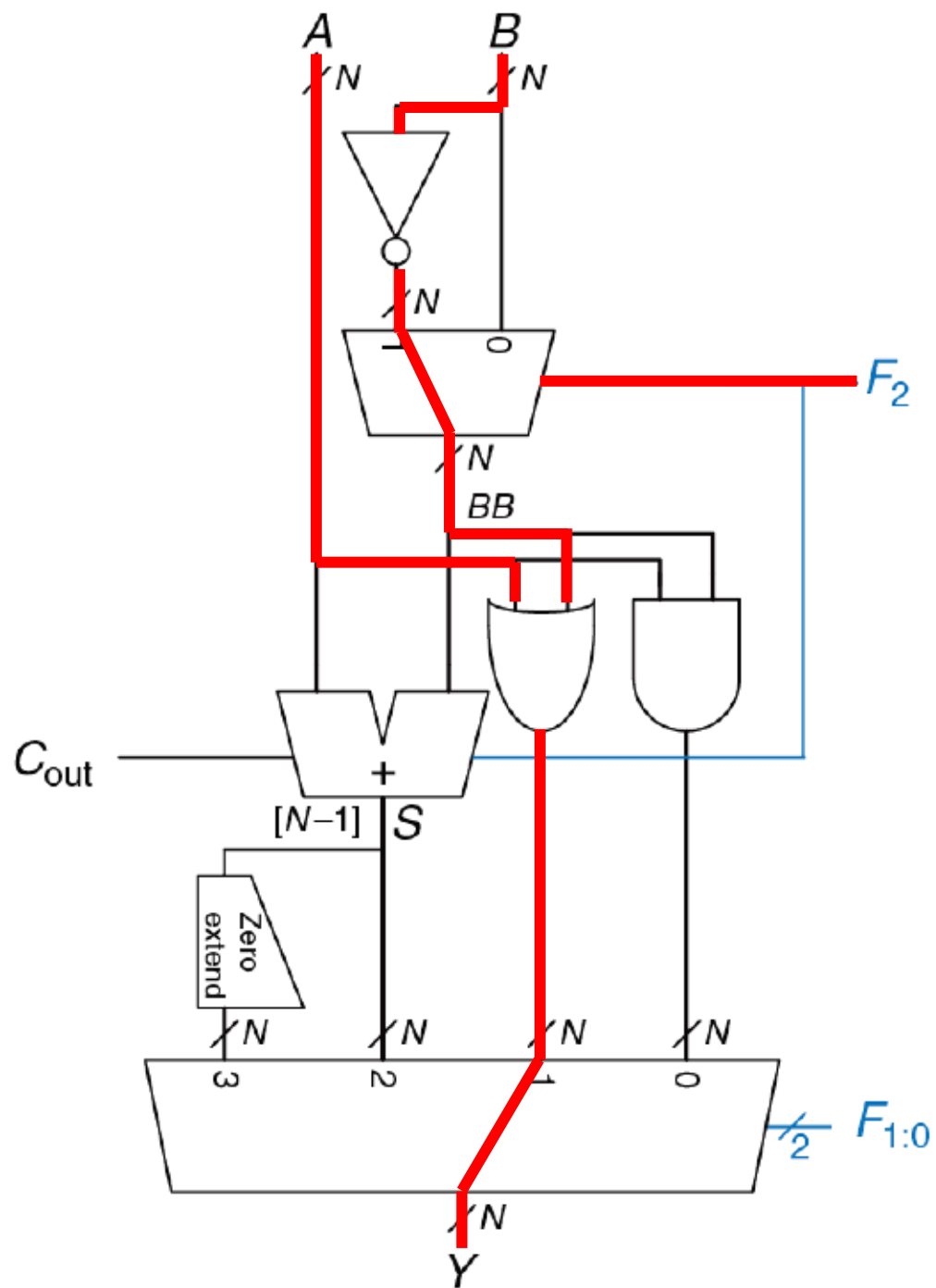




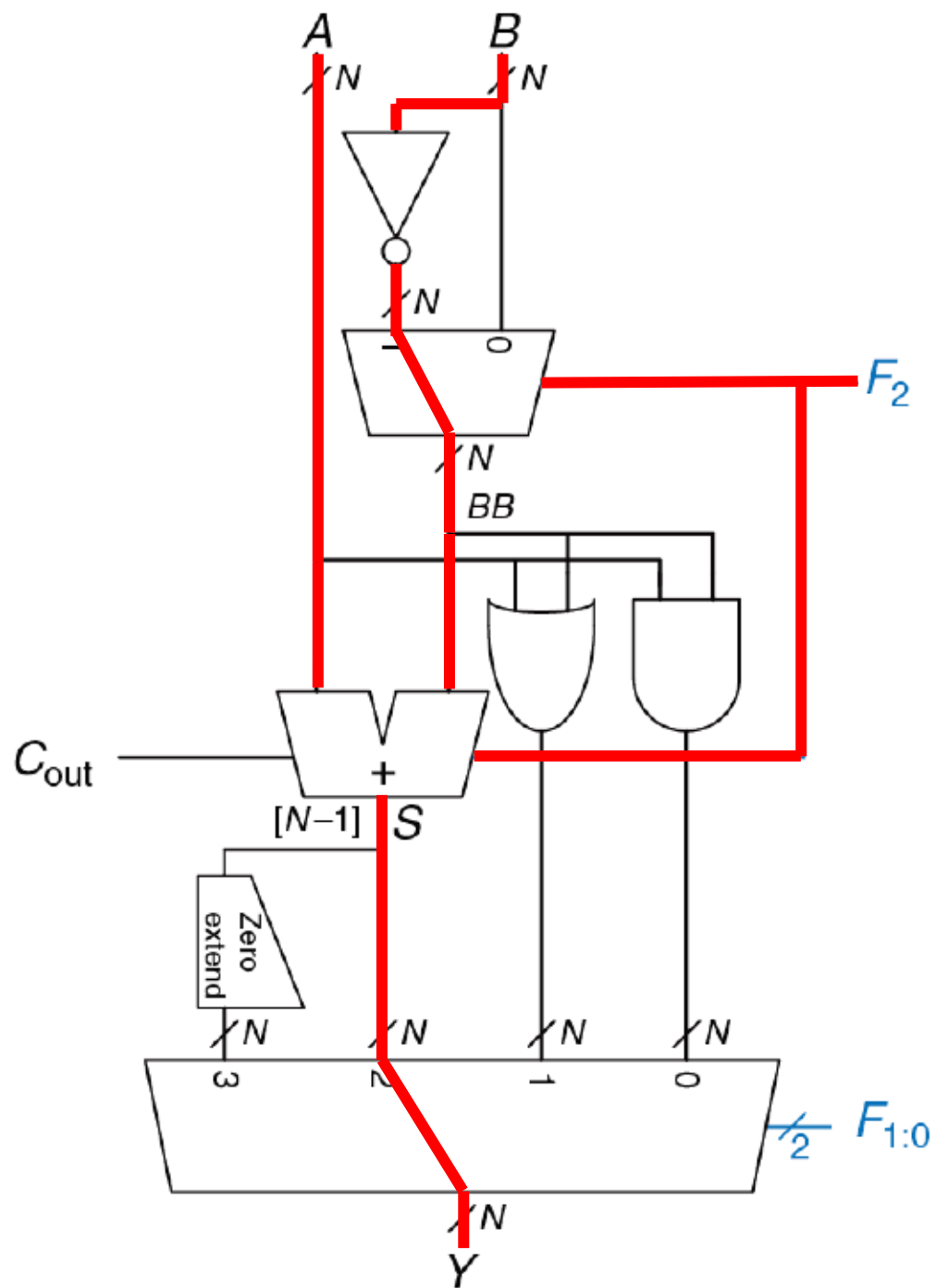




$F_{2:0}$	Function
000	$A \text{ AND } B$
001	$A \text{ OR } B$
010	$A + B$
011	not used
100	$A \text{ AND } \bar{B}$
101	$A \text{ OR } \bar{B}$
110	$A - B$
111	SLT



$F_{2:0}$	Function
000	A AND B
001	A OR B
010	A + B
011	not used
100	A AND $\bar{B}$
101	A OR $\bar{B}$
110	A - B
111	SLT



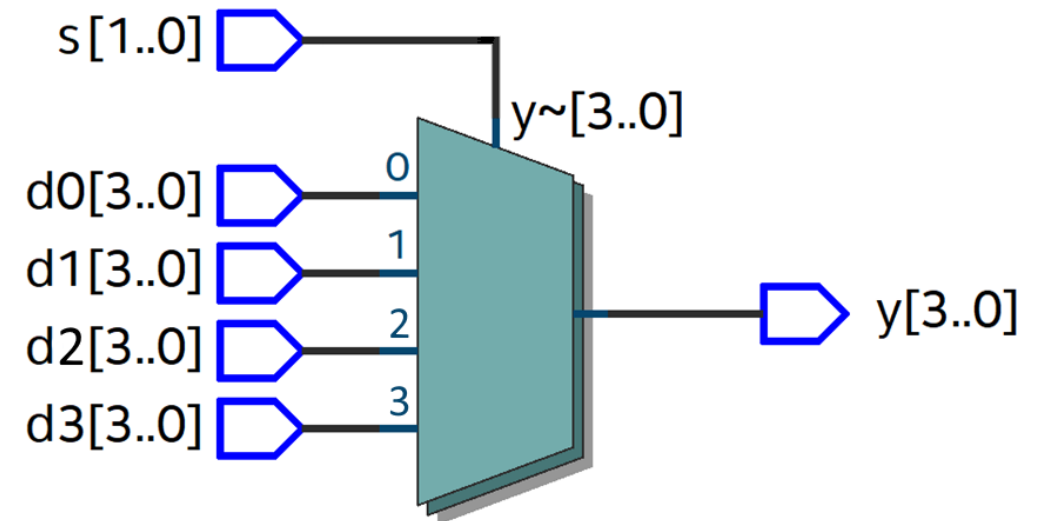
$F_{2:0}$	Function
000	A AND B
001	A OR B
010	A + B
011	not used
100	A AND $\bar{B}$
101	A OR $\bar{B}$
110	A - B
111	SLT





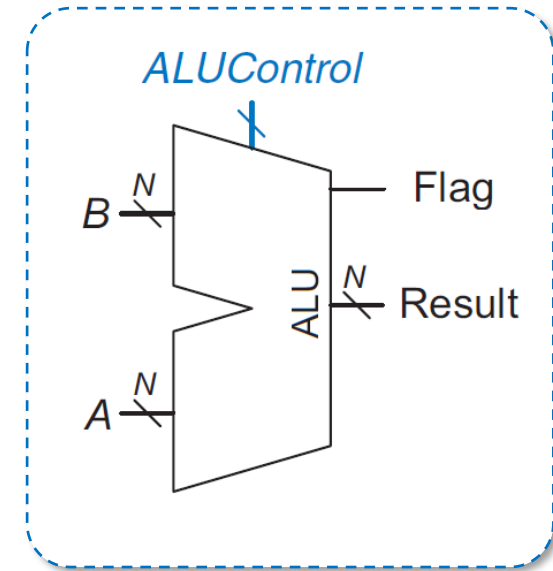
# CASE

```
module mux (  
    input [3:0] d0, d1, d2, d3,  
    input [1:0] s,  
    output reg [3:0] y  
);  
  
always @ (*) begin  
    case (s)  
        2'b00: y = d0;  
        2'b01: y = d1;  
        2'b10: y = d2;  
        2'b11: y = d3;  
    endcase  
end  
  
endmodule
```



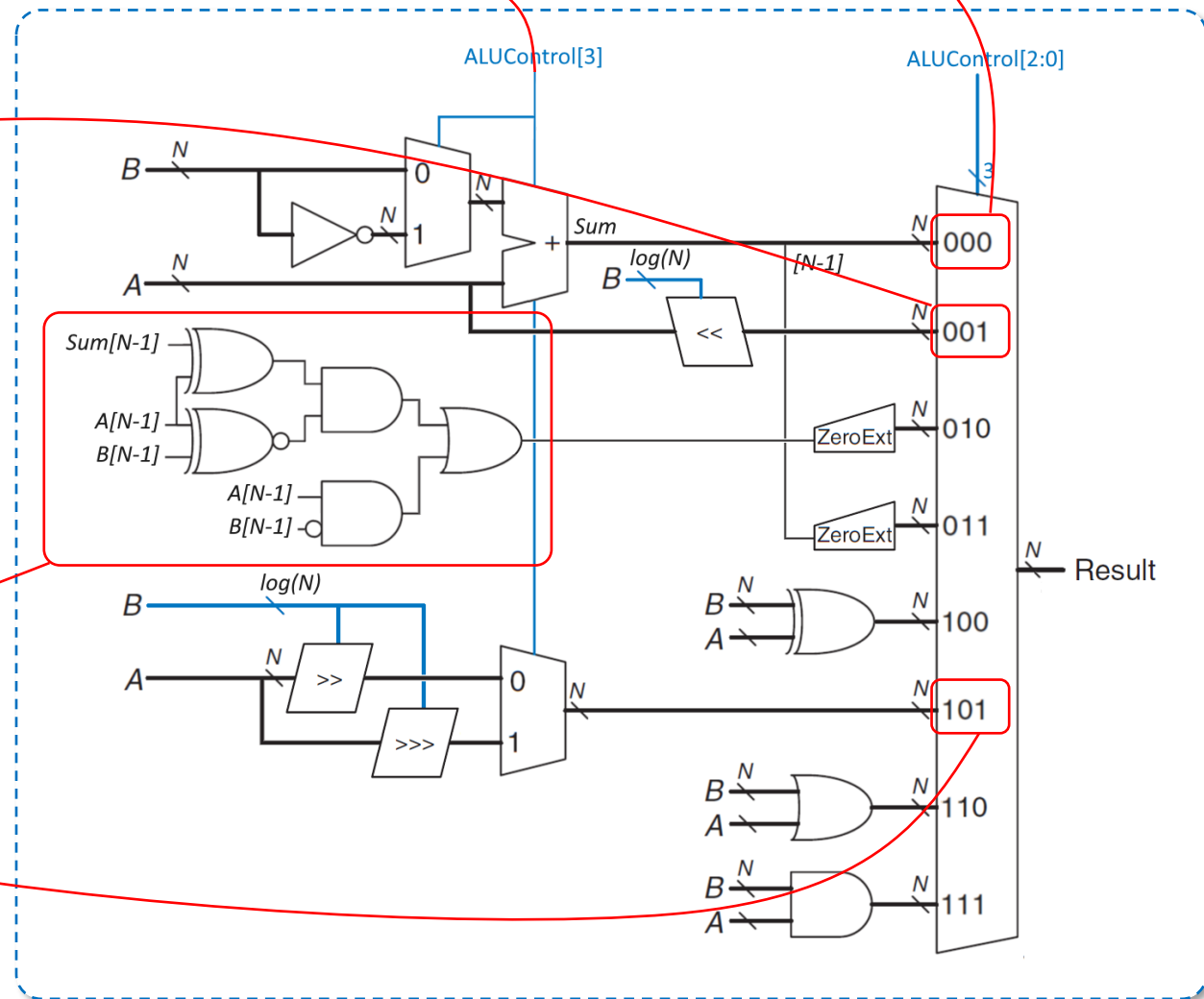
# ALU RISC-V

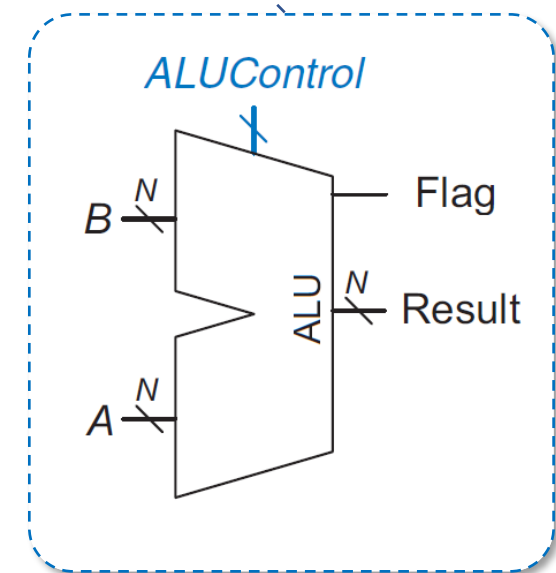
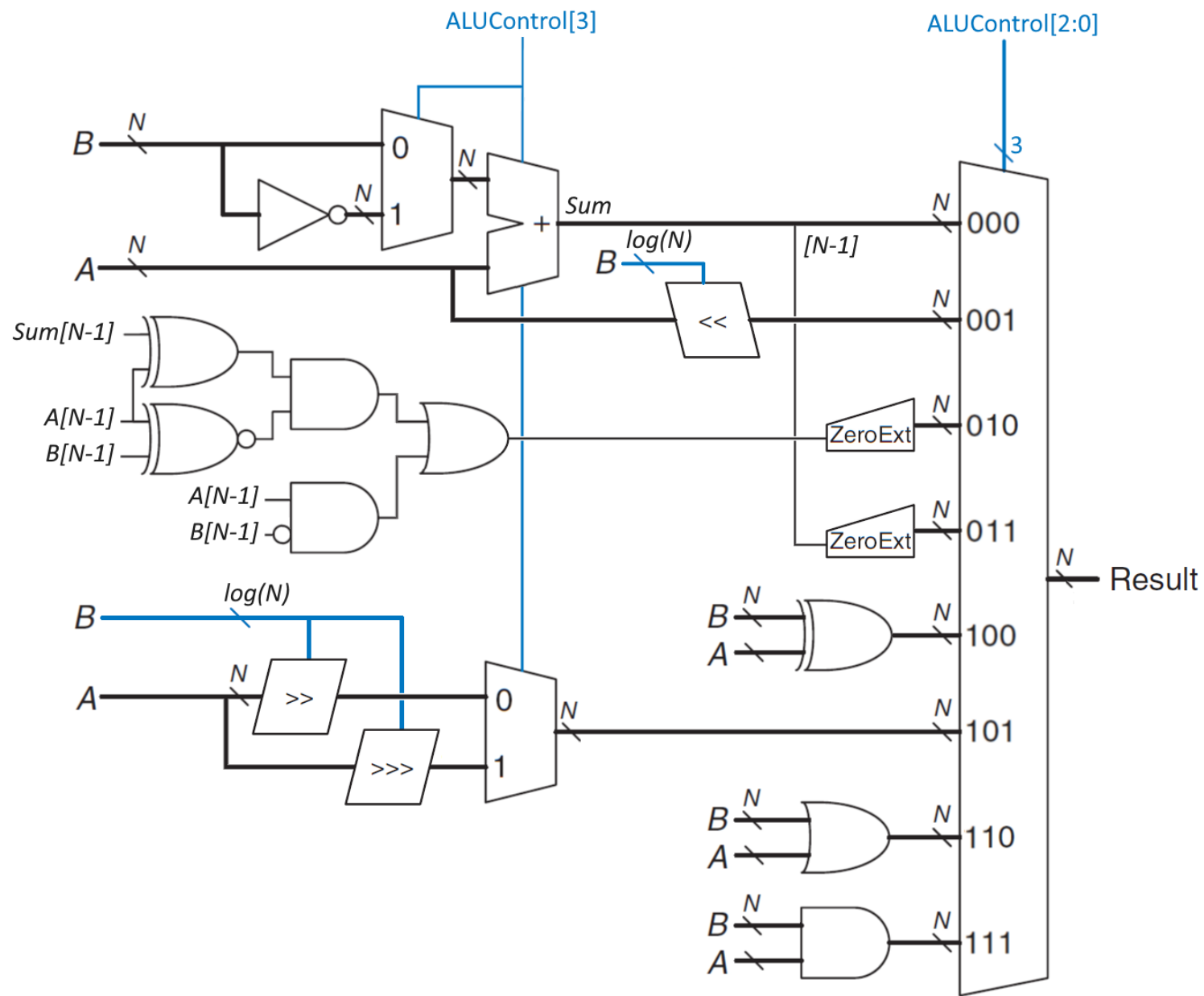
funct3	funct7	Description	Operation
000	0000000	add	$rd = rs1 + rs2$
000	0100000	sub	$rd = rs1 - rs2$
001	0000000	shift left logical	$rd = rs1 \ll rs2_{4:0}$
010	0000000	set less than	$rd = (rs1 < rs2)$
011	0000000	set less than unsigned	$rd = (rs1 < rs2)$
100	0000000	xor	$rd = rs1 \wedge rs2$
101	0000000	shift right logical	$rd = rs1 \gg rs2_{4:0}$
101	0100000	shift right arithmetic	$rd = rs1 \ggg rs2_{4:0}$
110	0000000	or	$rd = rs1 \mid rs2$
111	0000000	and	$rd = rs1 \& rs2$
000	-	branch if =	if $(rs1 == rs2)$ PC = BTA
001	-	branch if $\neq$	if $(rs1 \neq rs2)$ PC = BTA
100	-	branch if $<$	if $(rs1 < rs2)$ PC = BTA
101	-	branch if $\geq$	if $(rs1 \geq rs2)$ PC = BTA
110	-	branch if $<$ unsigned	if $(rs1 < rs2)$ PC = BTA
111	-	branch if $\geq$ unsigned	if $(rs1 \geq rs2)$ PC = BTA

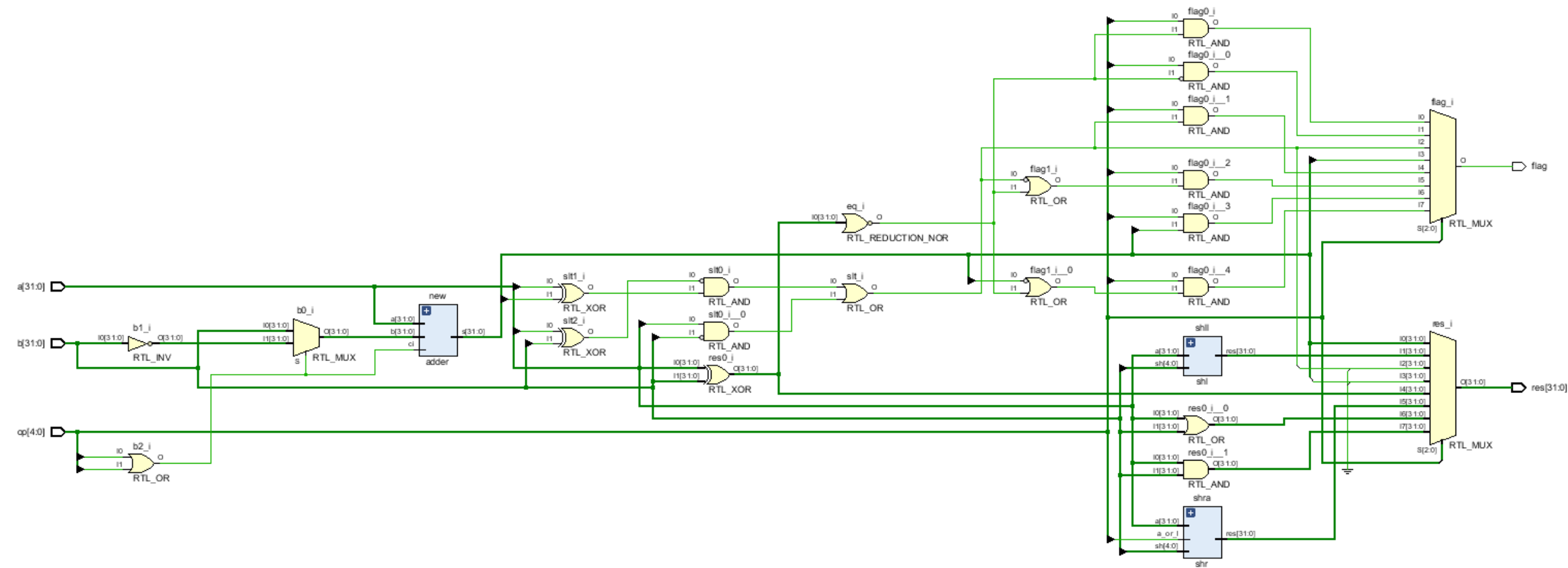




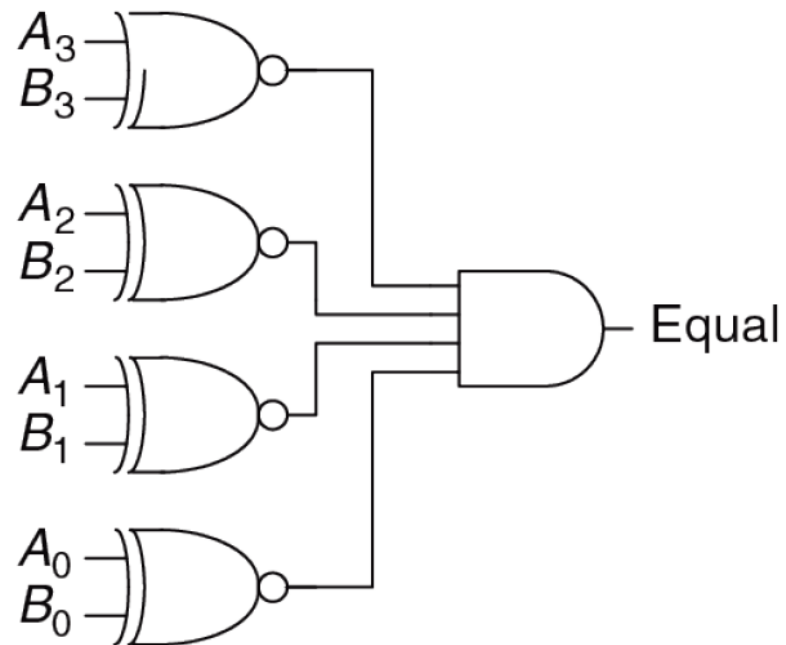
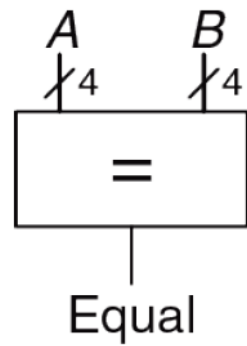
func3	func7	Description	Operation
000	0000000	add	rd = rs1 + rs2
000	0100000	sub	rd = rs1 - rs2
001	0000000	shift left logical	rd = rs1 << rs2 <sub>4:0</sub>
010	0000000	set less than	rd = (rs1 < rs2)
011	0000000	set less than unsigned	rd = (rs1 < rs2)
100	0000000	xor	rd = rs1 ^ rs2
101	0000000	shift right logical	rd = rs1 >> rs2 <sub>4:0</sub>
101	0100000	shift right arithmetic	rd = rs1 >>> rs2 <sub>4:0</sub>
110	0000000	or	rd = rs1   rs2
111	0000000	and	rd = rs1 & rs2
000	-	branch if =	if (rs1 == rs2) PC = BTA
001	-	branch if ≠	if (rs1 ≠ rs2) PC = BTA
100	-	branch if <	if (rs1 < rs2) PC = BTA
101	-	branch if ≥	if (rs1 ≥ rs2) PC = BTA
110	-	branch if < unsigned	if (rs1 < rs2) PC = BTA
111	-	branch if ≥ unsigned	if (rs1 ≥ rs2) PC = BTA



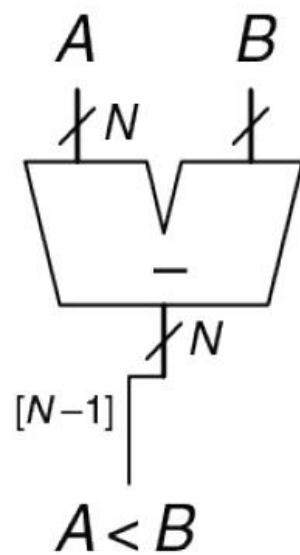




# Равенство

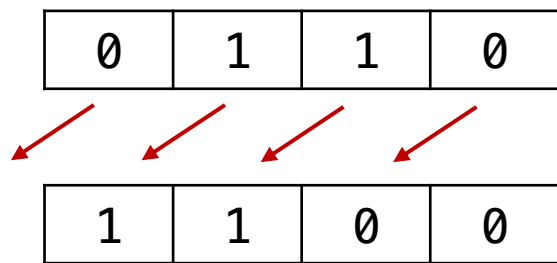


# Сравнение

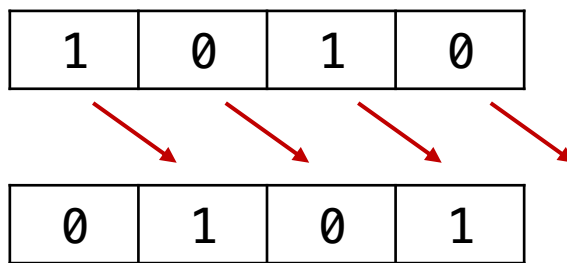


# Сдвиг

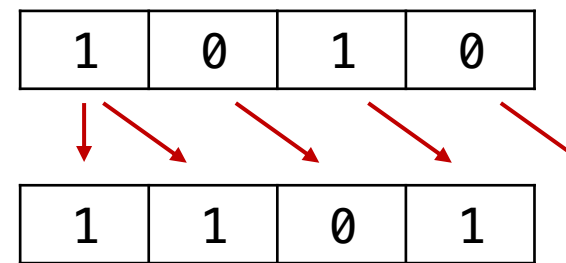
<<

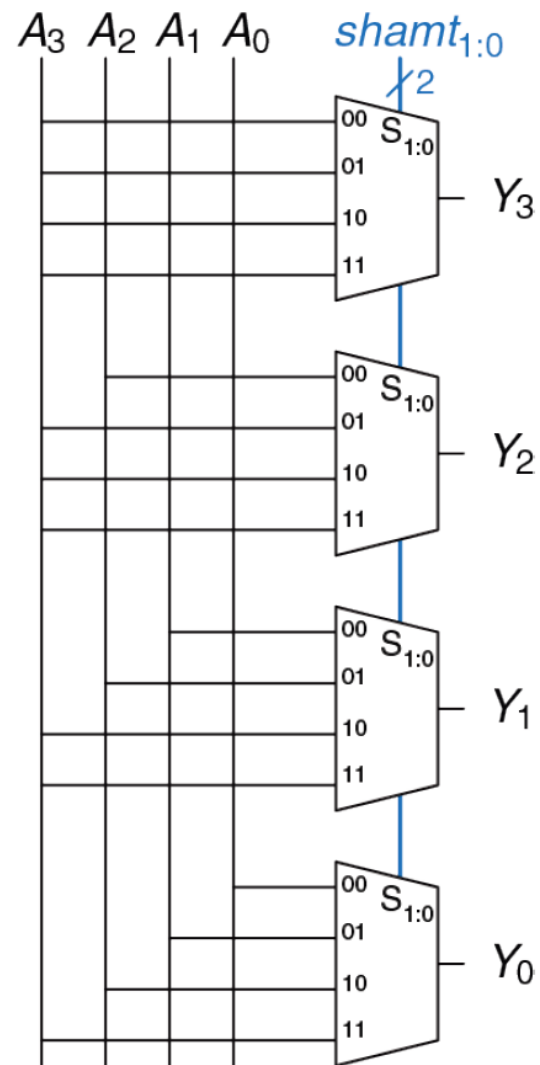
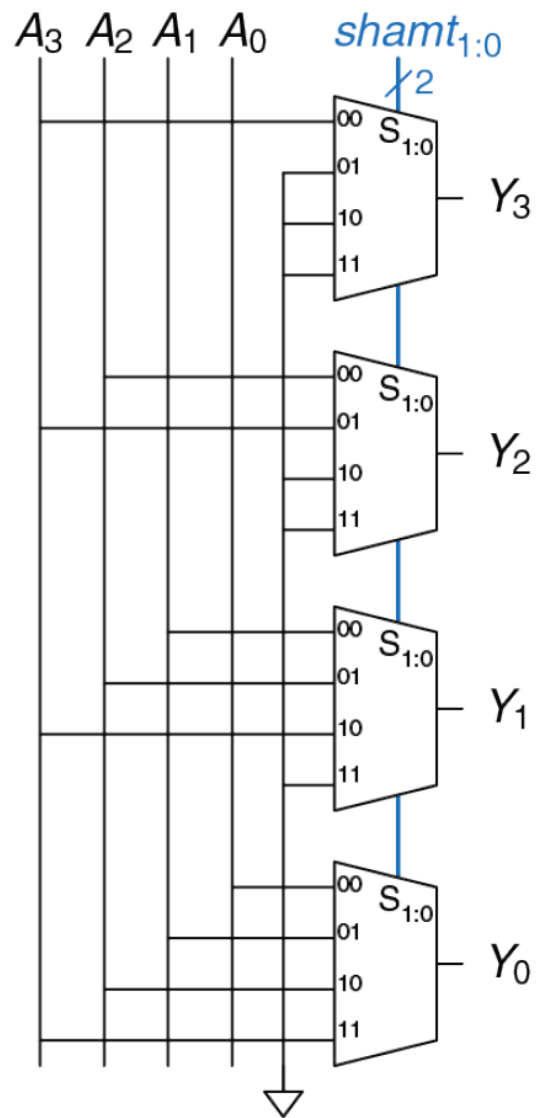
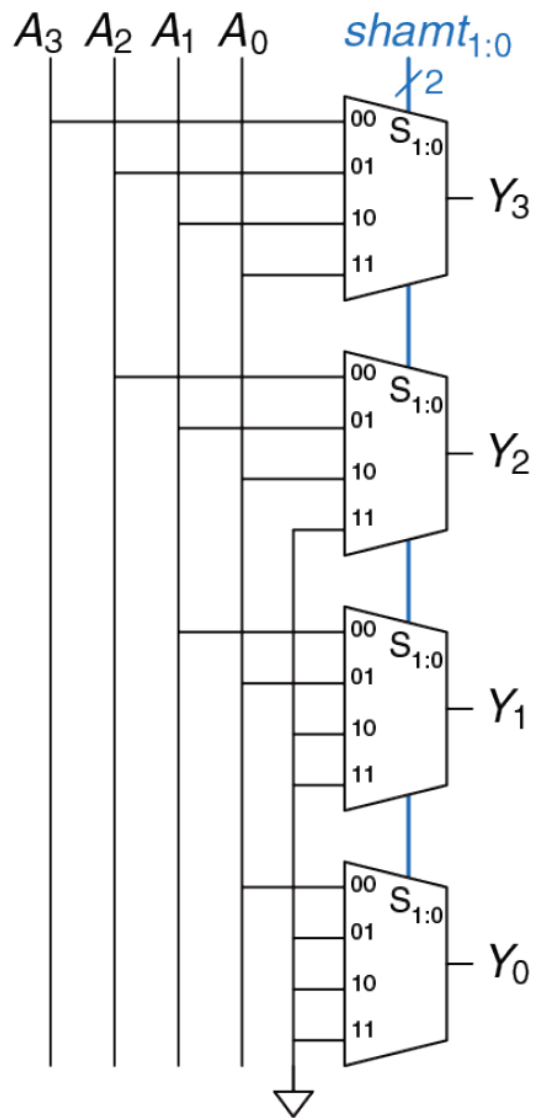
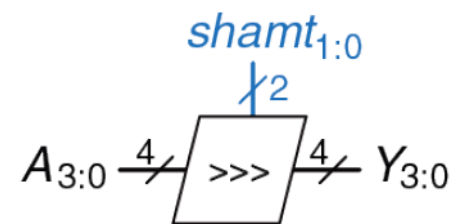
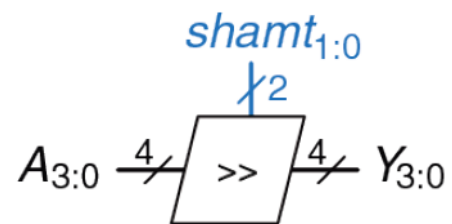
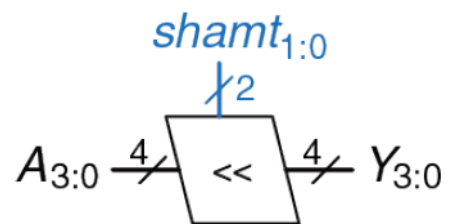


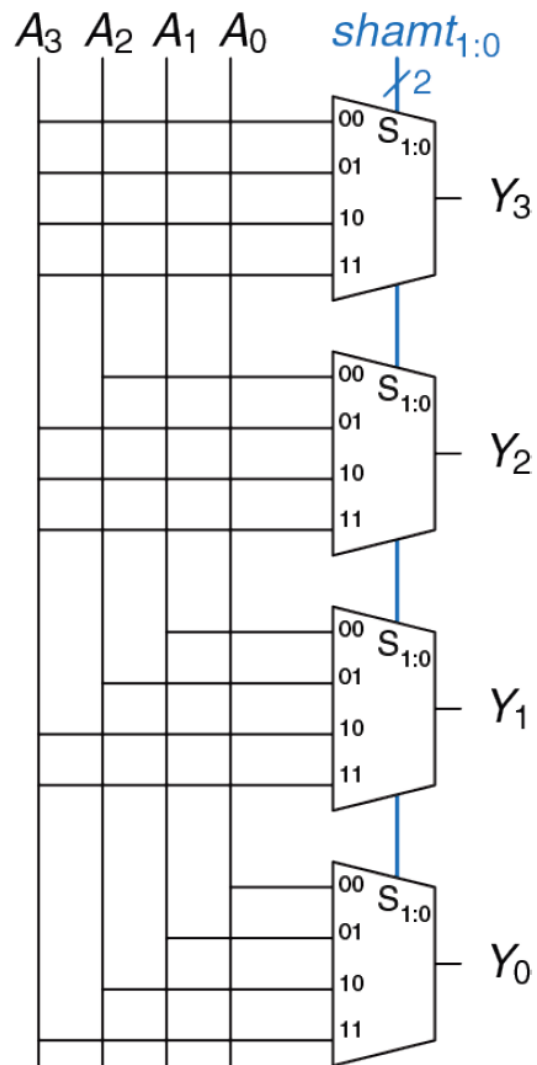
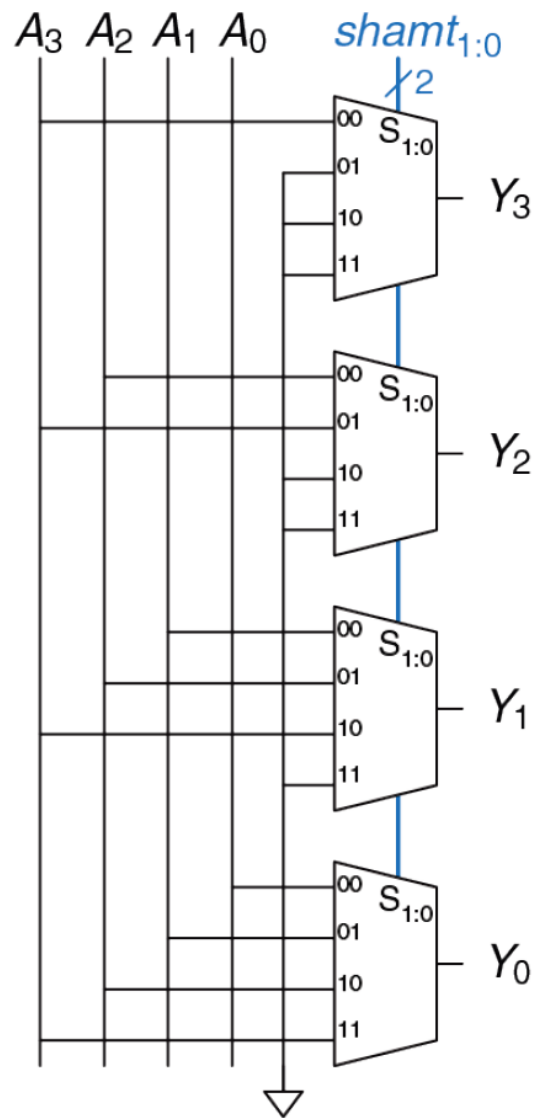
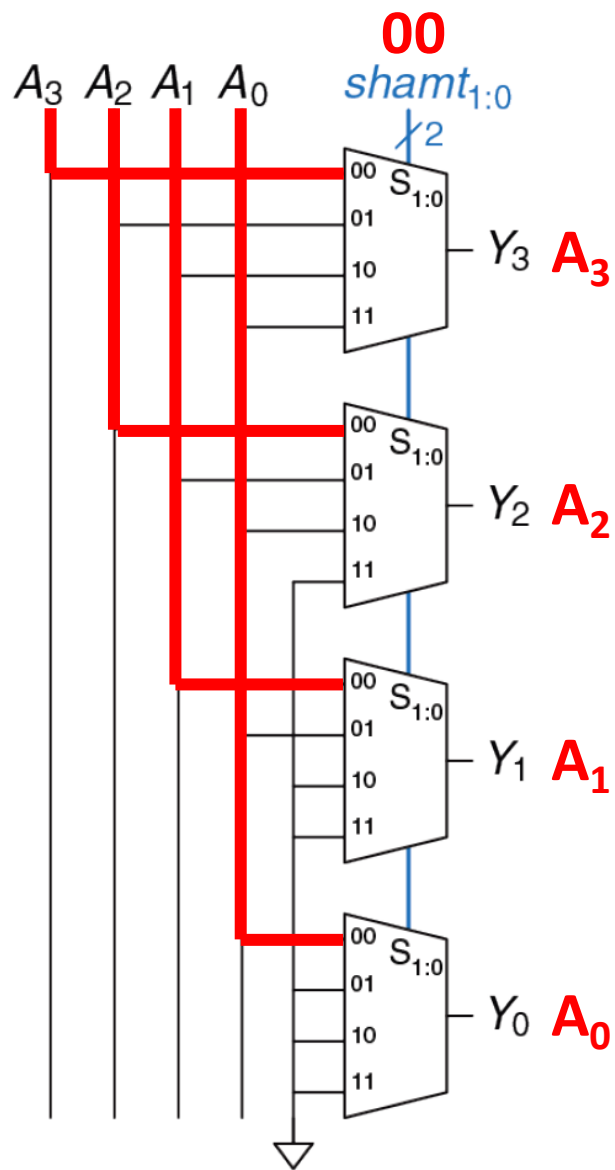
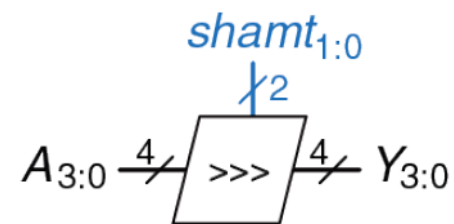
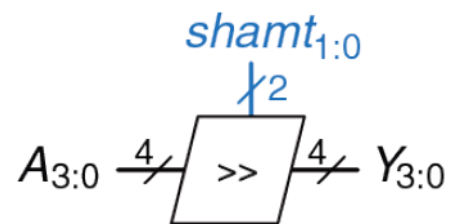
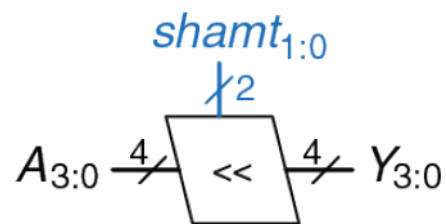
>>



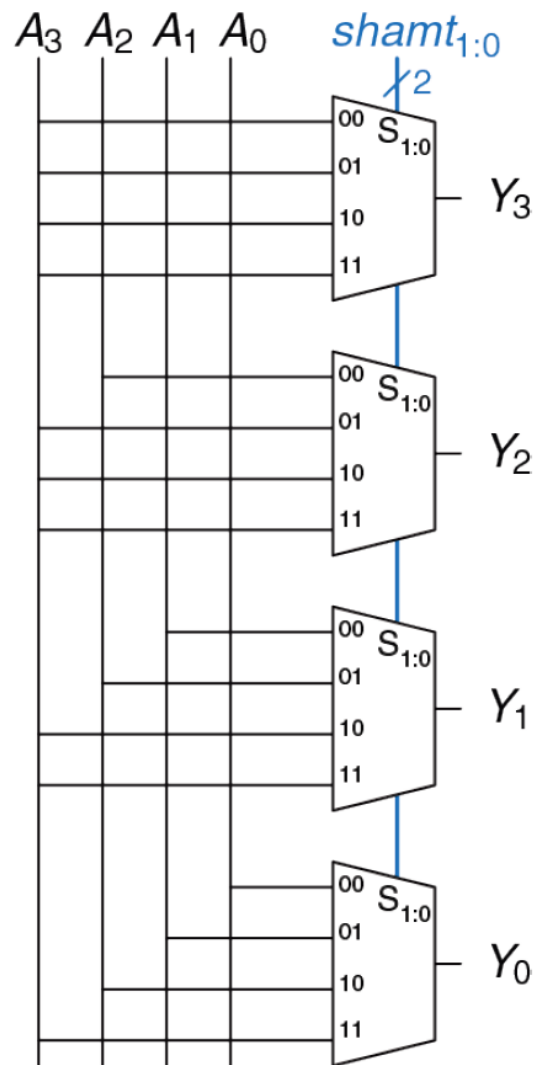
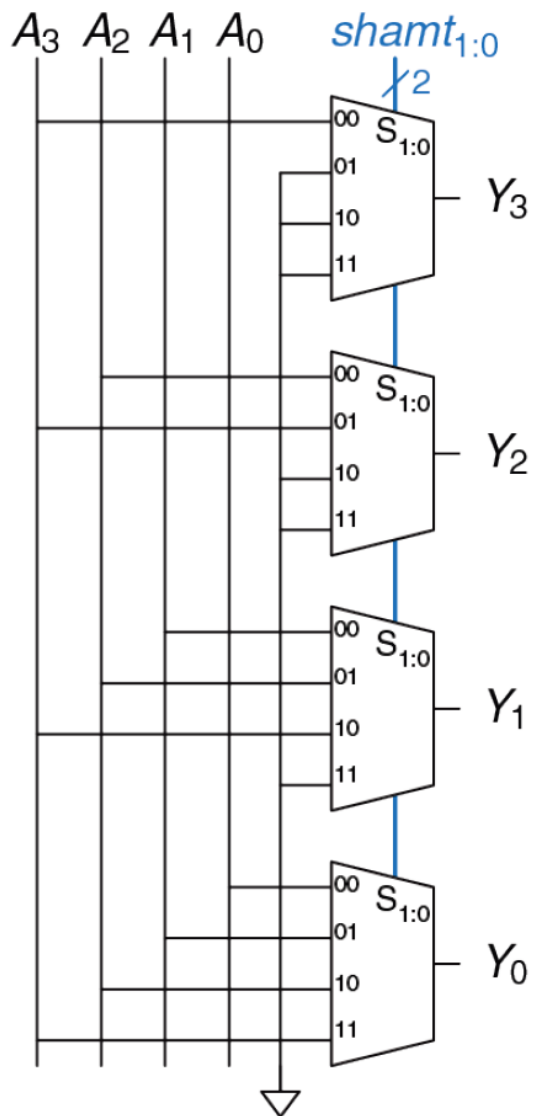
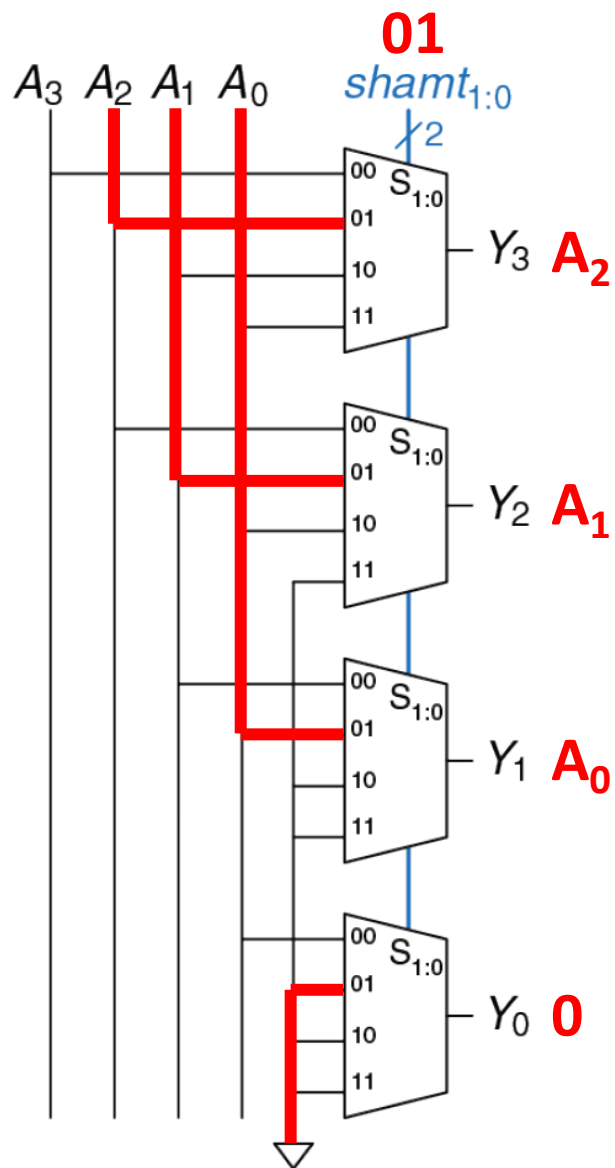
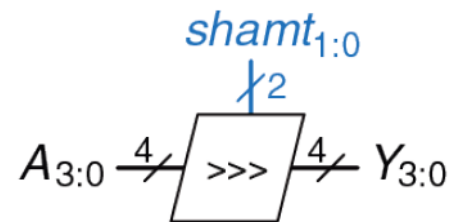
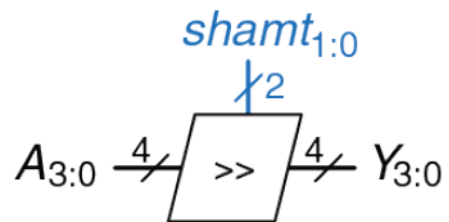
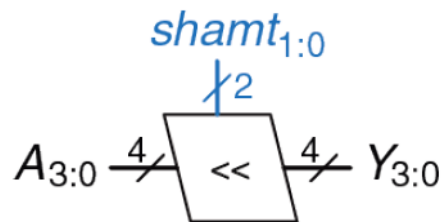
>>>

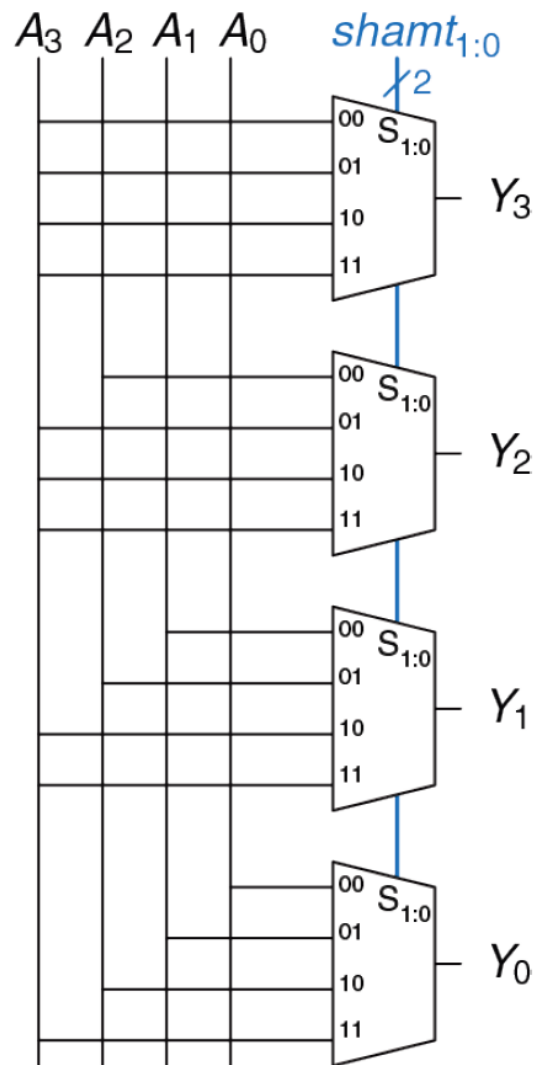
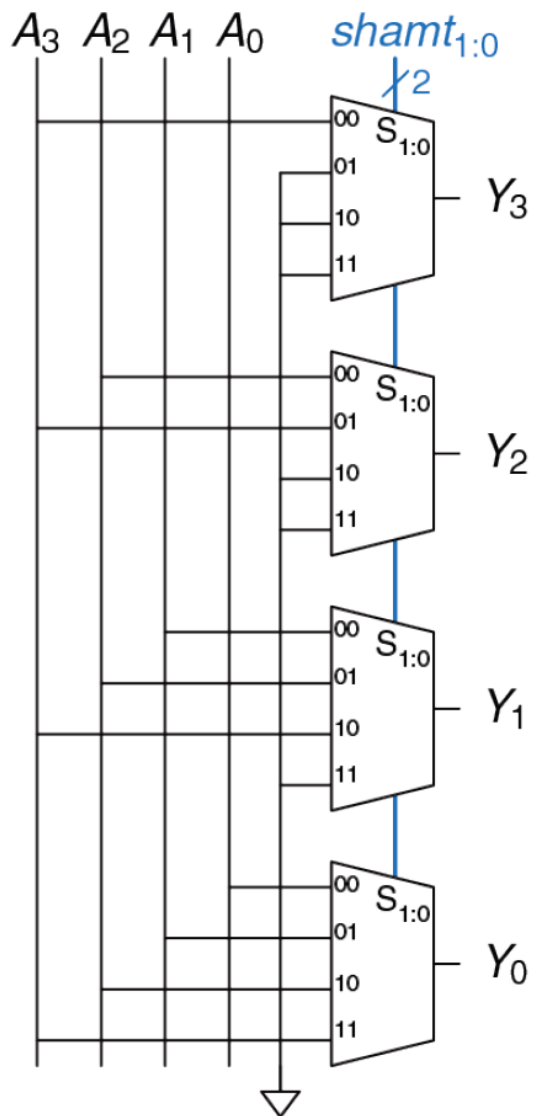
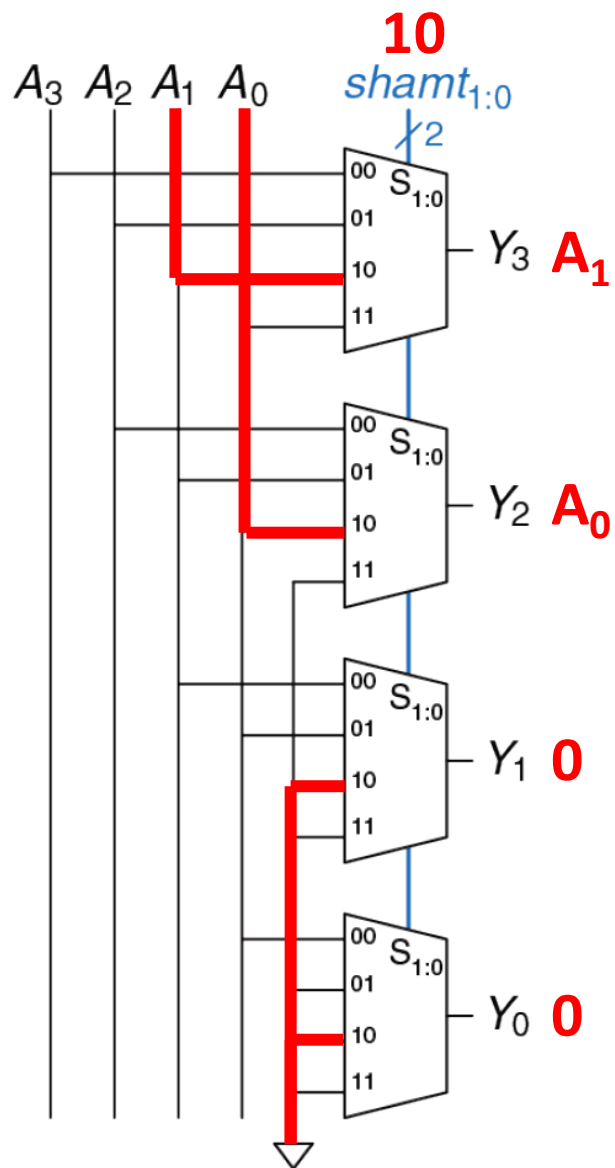
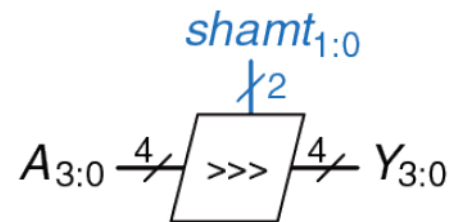
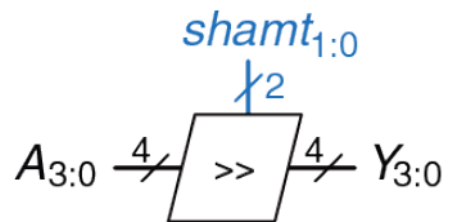
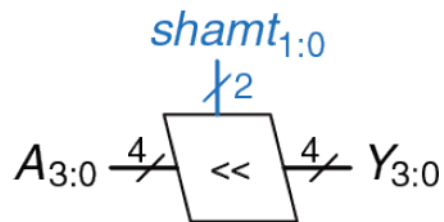


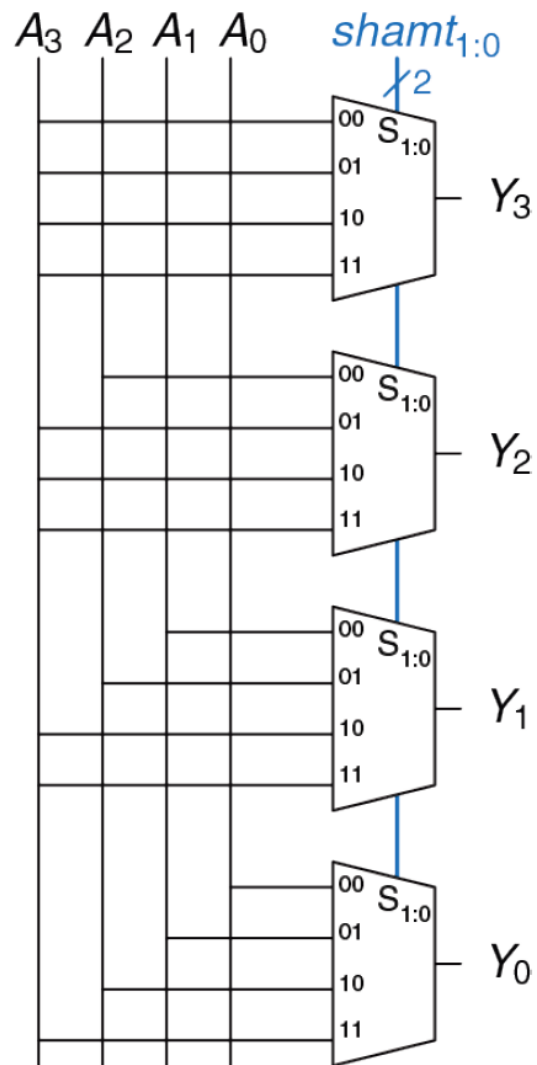
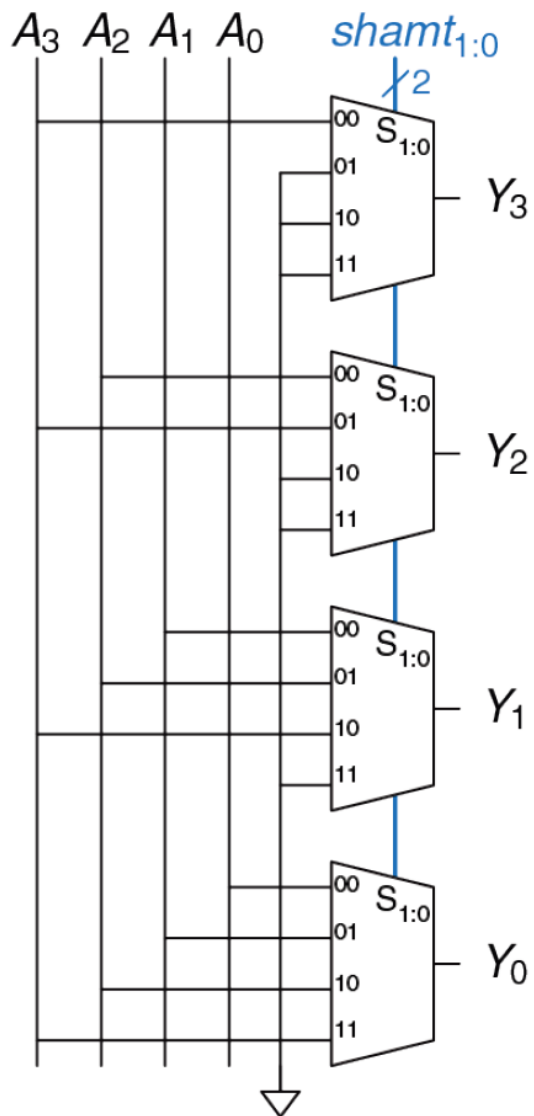
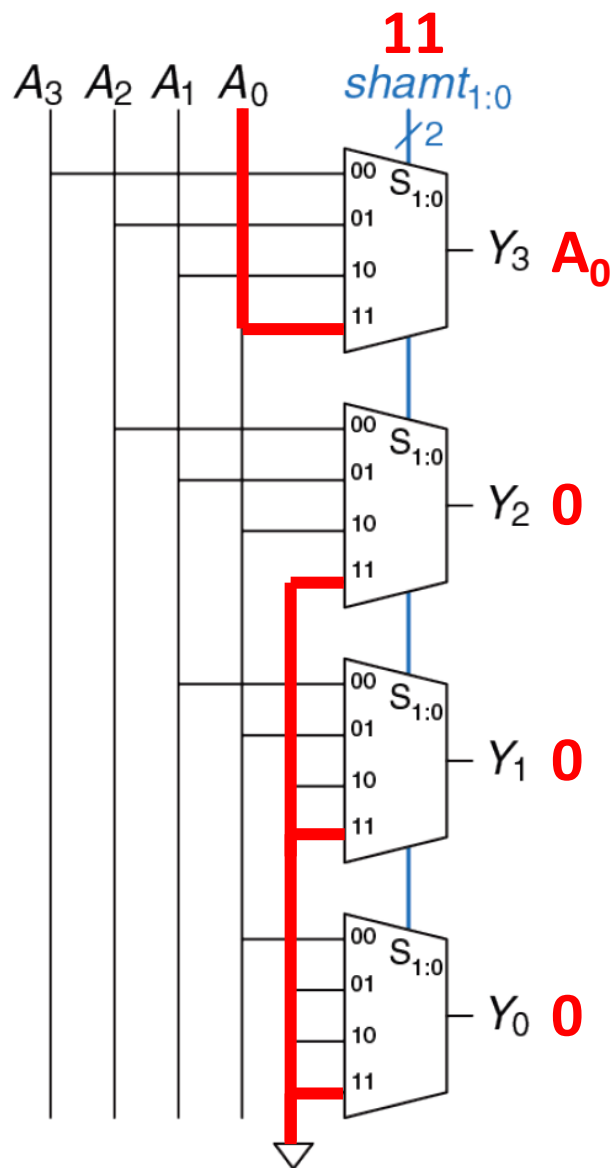
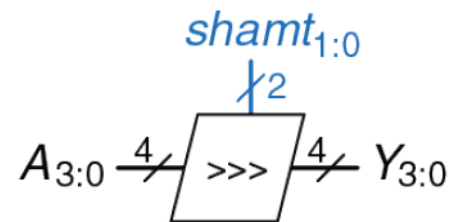
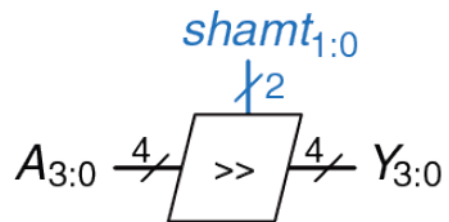
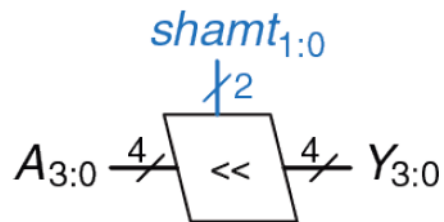


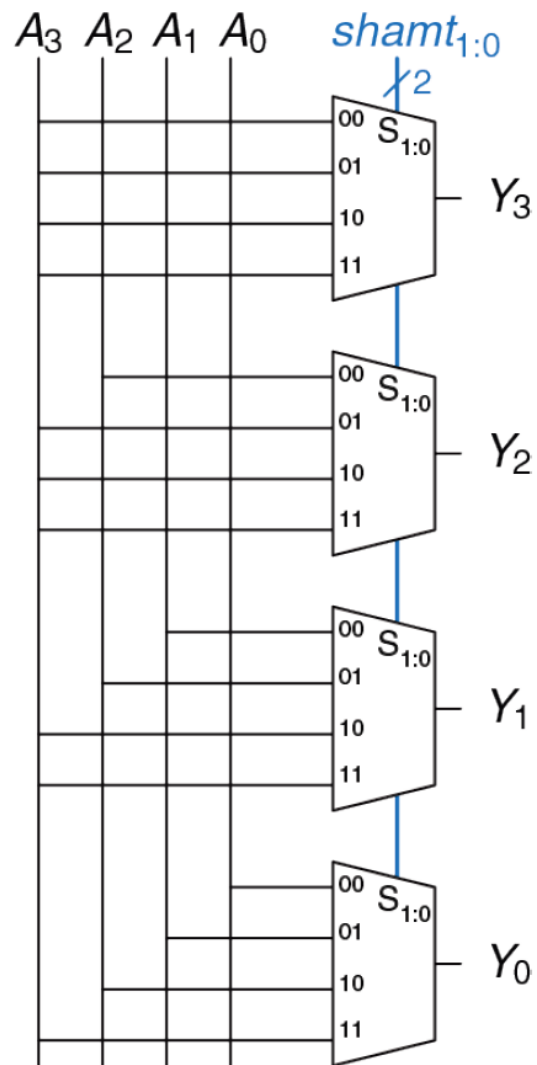
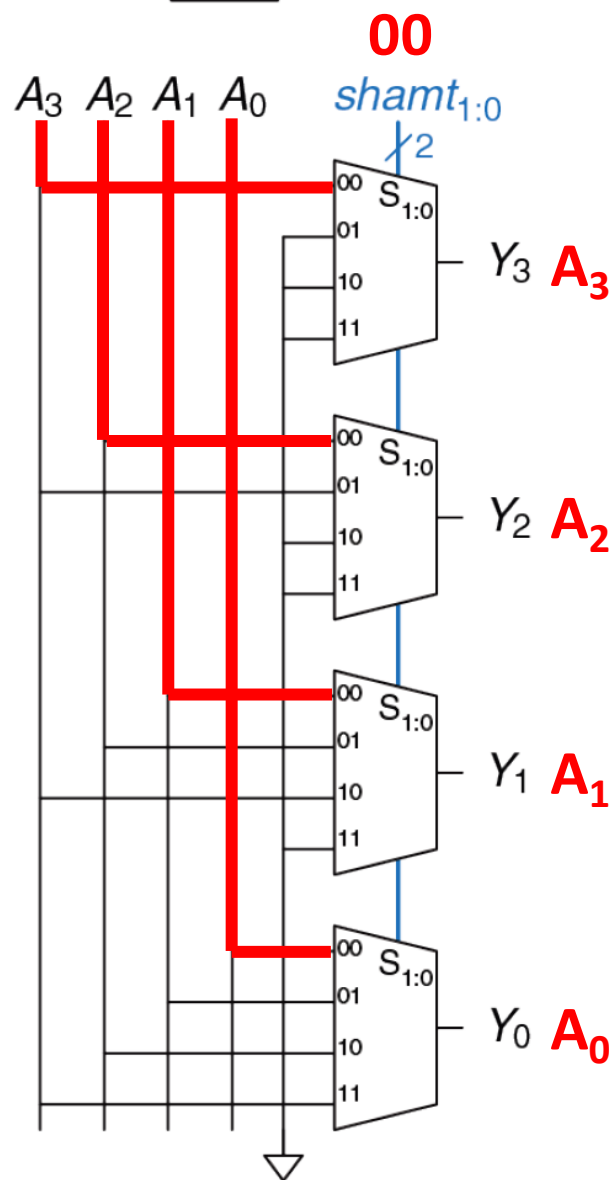
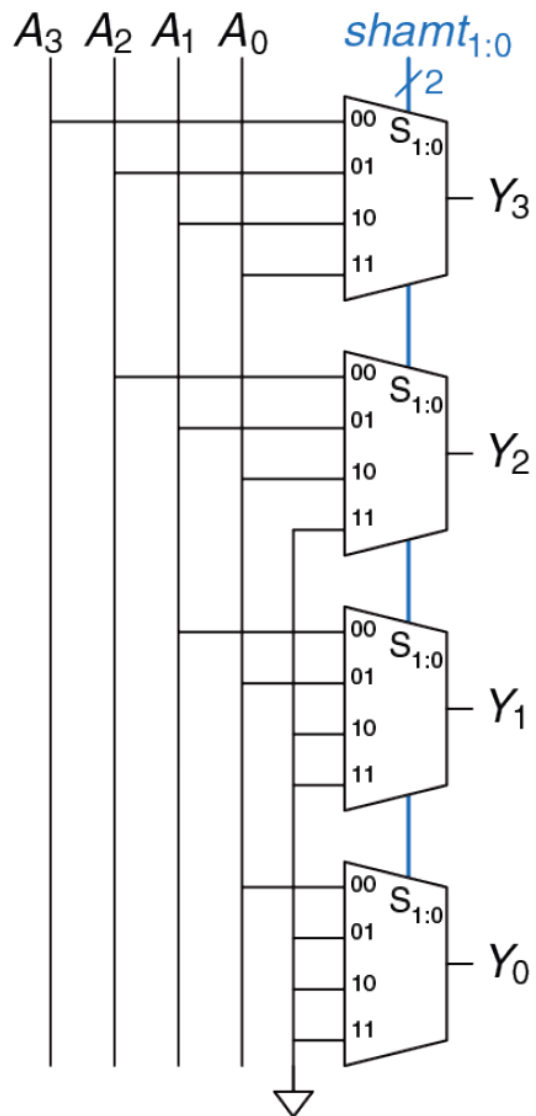
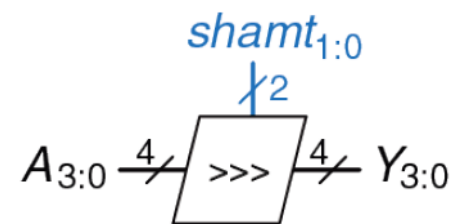
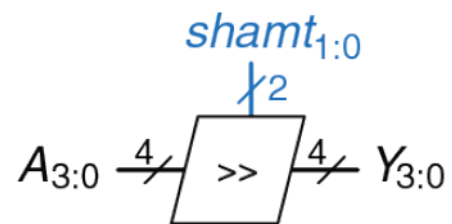
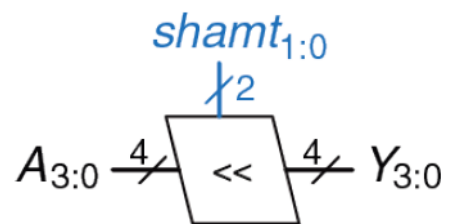


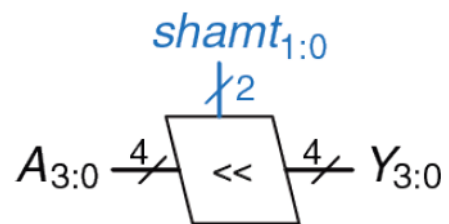


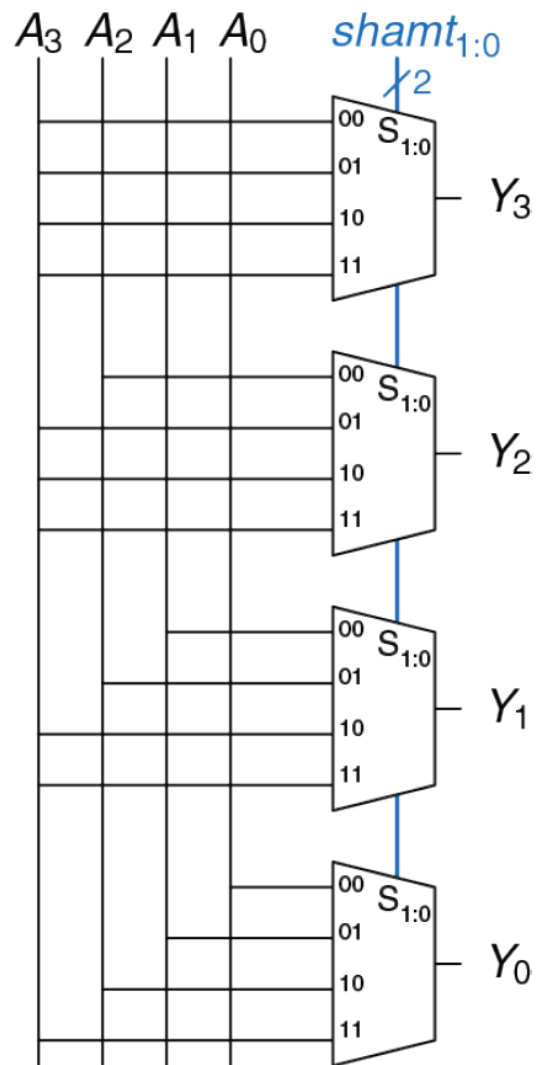
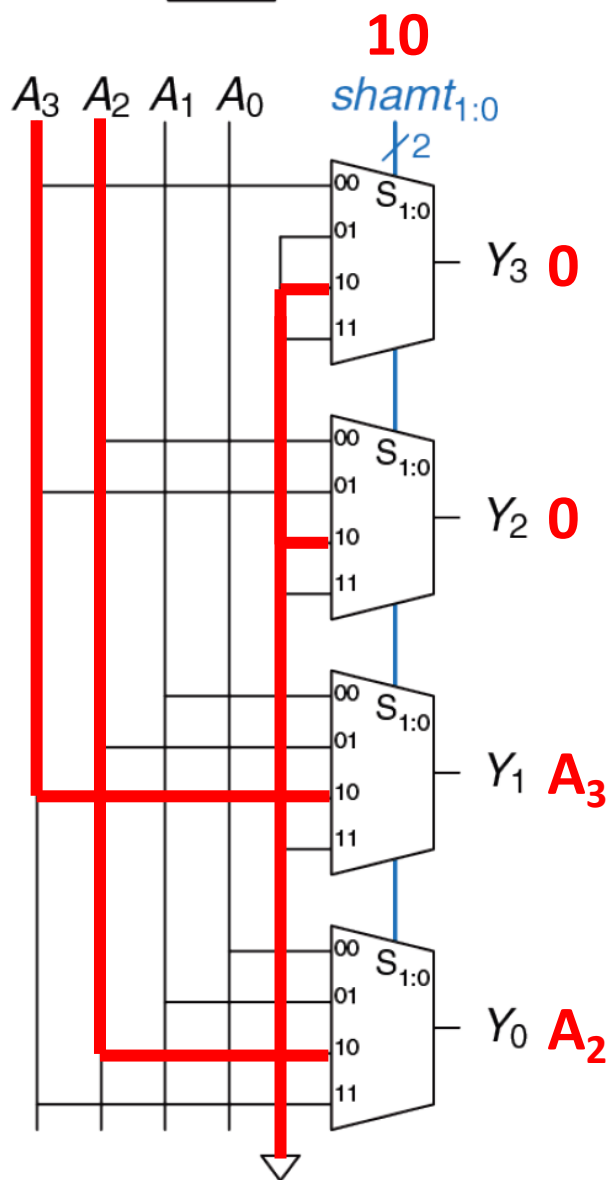
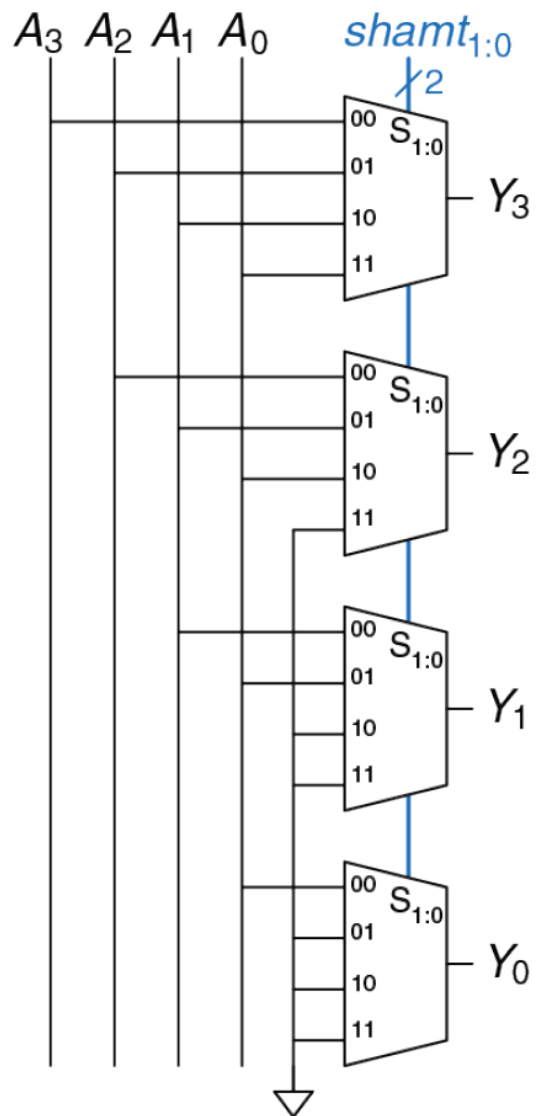
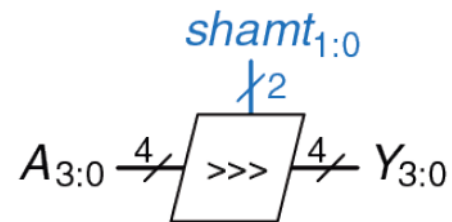
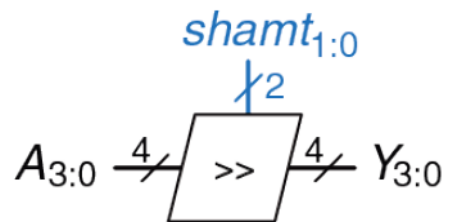
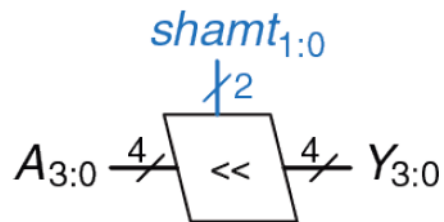


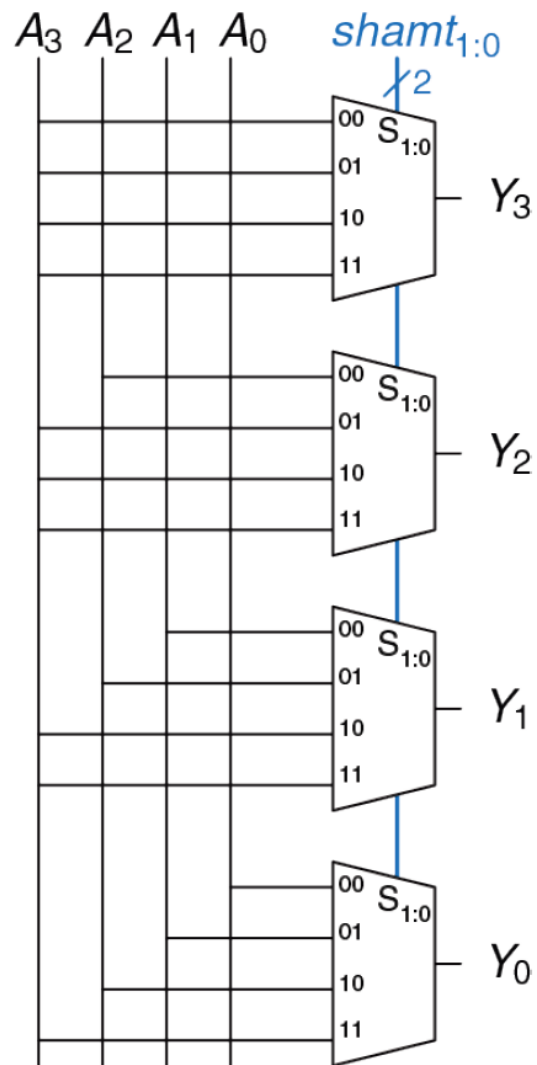
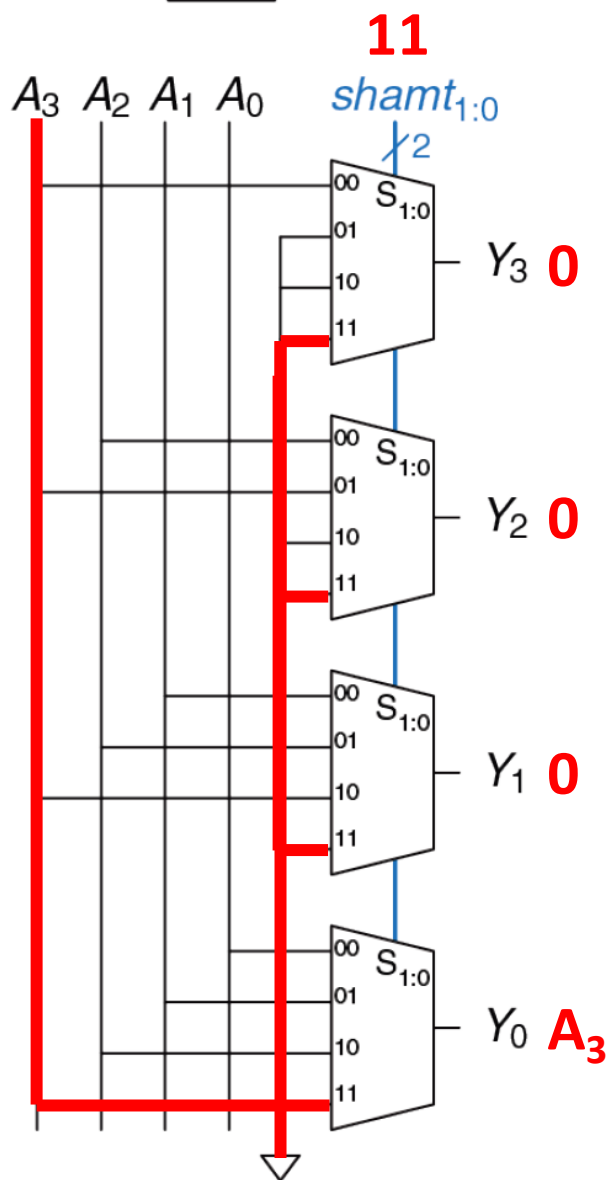
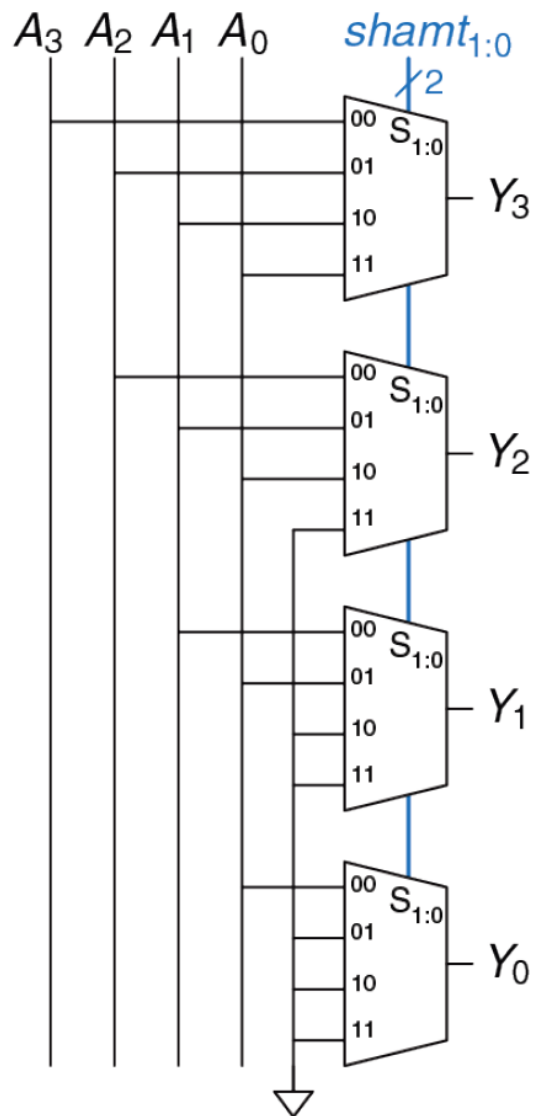
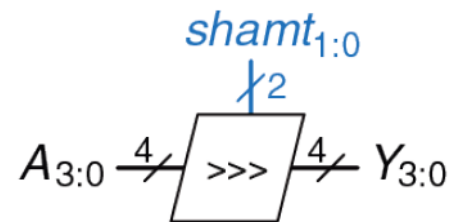
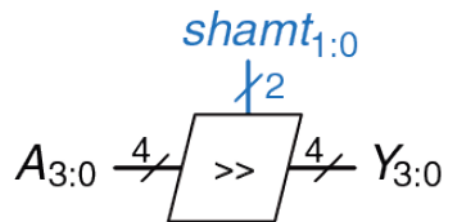
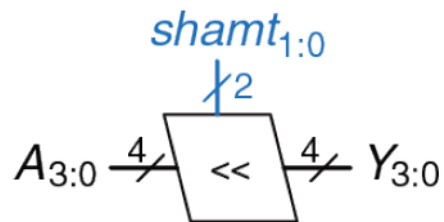


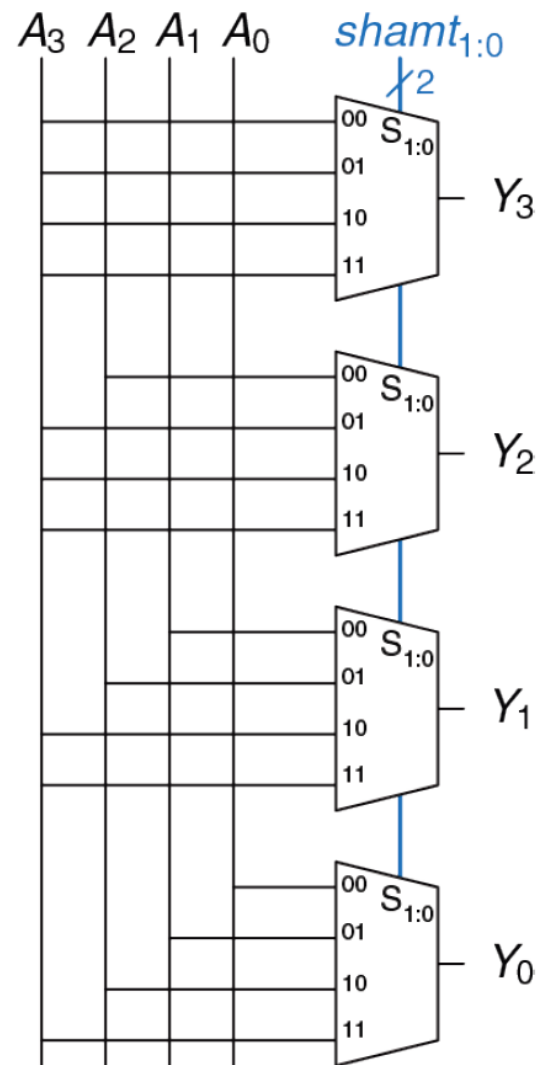
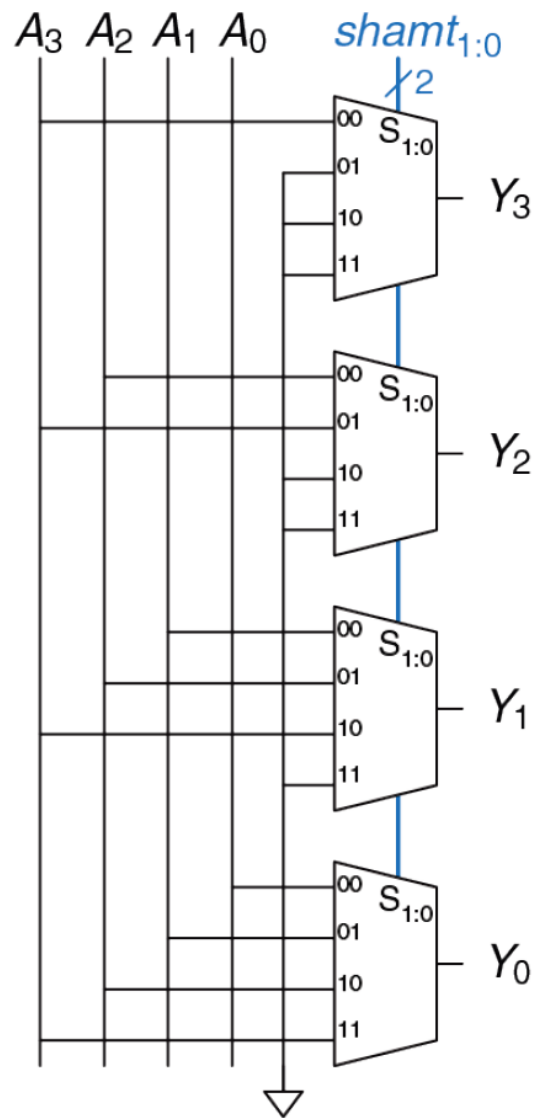
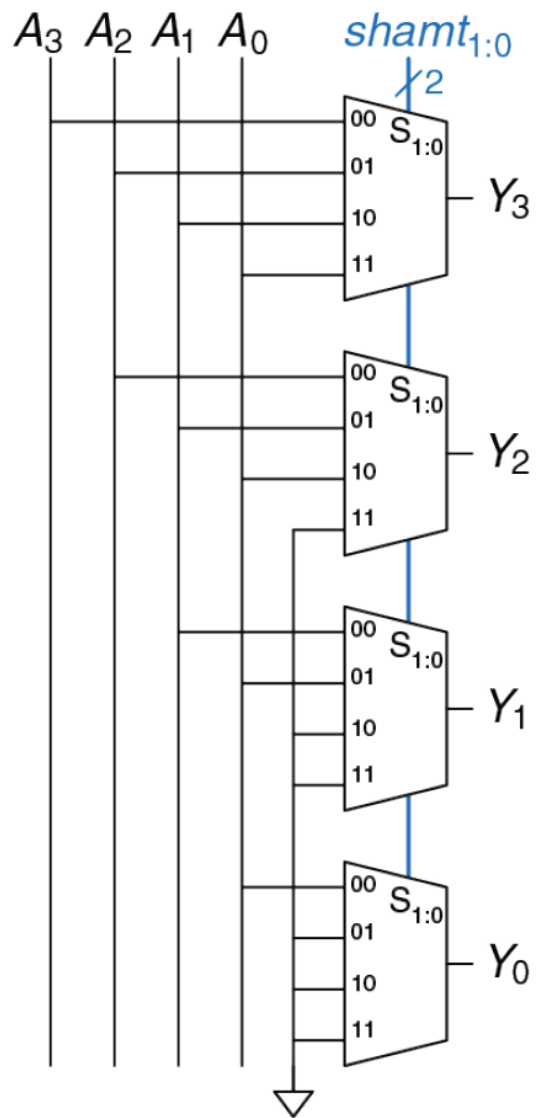
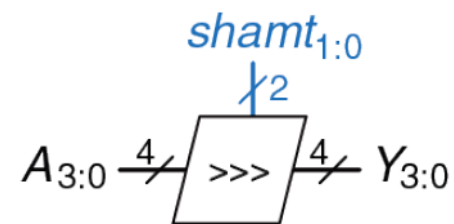
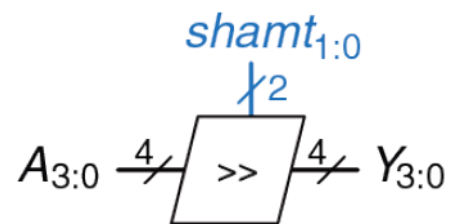
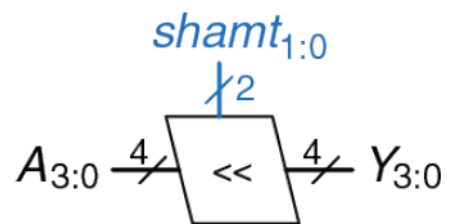














# Альтернативное АЛУ

$$R_i^+ = a_i \oplus b_i \oplus MP_{i-1}$$

$$R_i^+ = (\overline{a_i} \vee \overline{b_i})(a_i \vee b_i) \oplus MP_{i-1} = \overline{a_i b_i}(a_i \vee b_i) \oplus MP_{i-1}$$

$$R_i' = \overline{S_3 a_i b_i}(a_i \vee b_i) \oplus MP_{i-1}$$

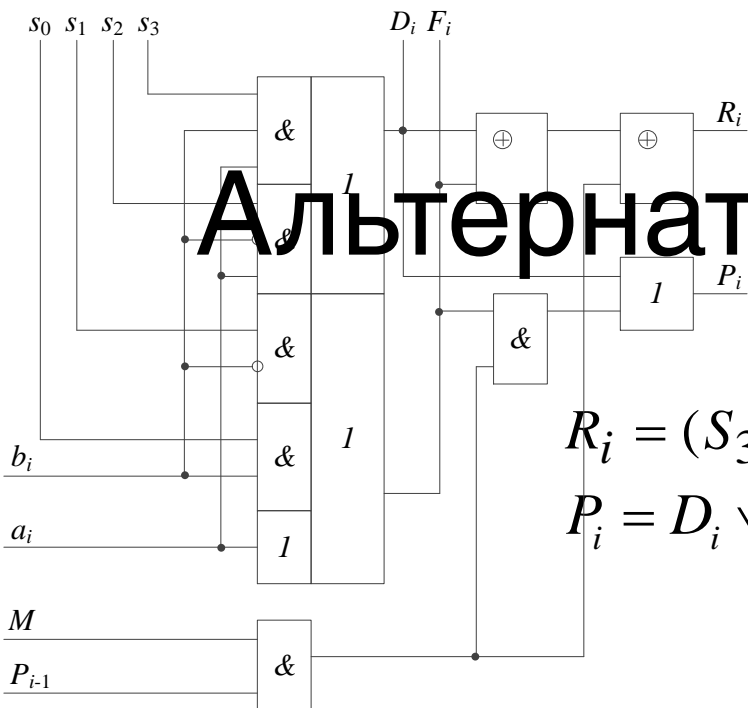
$$R_i^- = a_i \oplus \overline{b_i} \oplus MP_{i-1}$$

$$R_i^- = (\overline{a_i} \vee b_i)(a_i \vee \overline{b_i}) \oplus MP_{i-1} = \overline{a_i \overline{b_i}}(a_i \vee \overline{b_i}) \oplus MP_{i-1}$$

$$R_i'' = \overline{S_2 a_i \overline{b_i}}(a_i \vee \overline{b_i}) \oplus MP_{i-1}$$

$$R_i = \overline{S_3 a_i b_i \vee S_2 a_i \overline{b_i}} \cdot (a_i \vee S_1 \overline{b_i} \vee S_0 b_i) \oplus MP_{i-1}$$

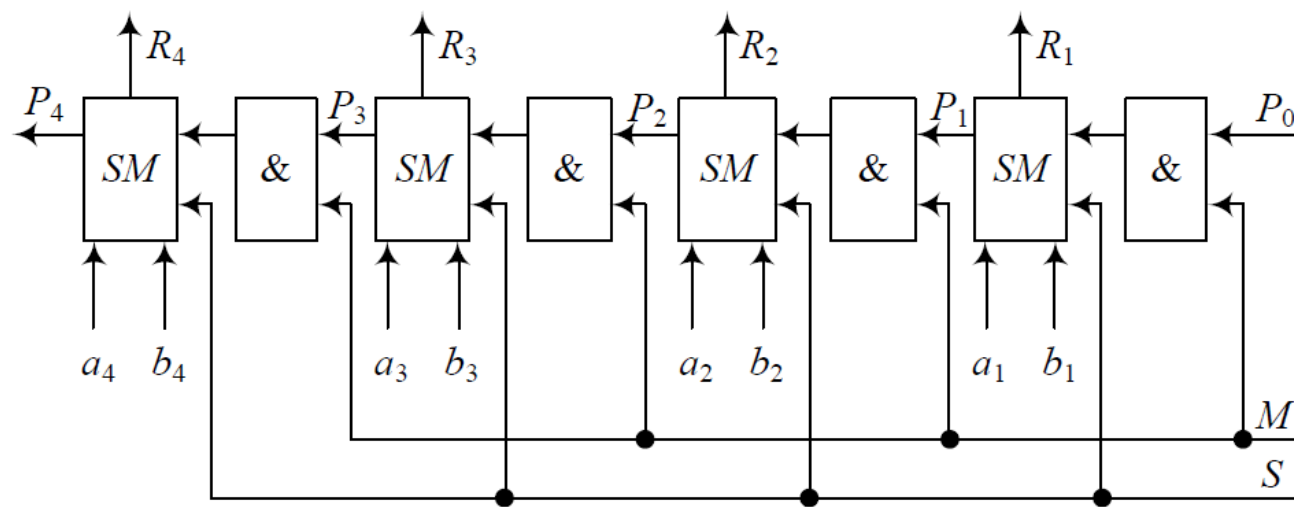
# Альтернативное АЛУ



$$R_i = (S_3 \cdot a_i \cdot b_i \vee S_2 \cdot a_i \cdot \bar{b}_i) \oplus (a_i \vee S_1 \cdot \bar{b}_i \vee S_0 \cdot b_i) \oplus M P_{i-1}$$

$$P_i = D_i \vee F_i P_{i-1}.$$

# Альтернативное АЛУ



# Альтернативное АЛУ

№ п/п	Управляющее слово				$R$		
					При $M = 0$	При $M = 1$	
	$S_3$	$S_2$	$S_1$	$S_0$		При $P_0 = 0$	При $P_0 = 1$
1	0	0	0	0	$A$	$A$	$A + 1$
2	0	0	0	1	$A \vee B$	$A \vee B$	$A \vee B + 1$
3	0	0	1	0	$A \vee \overline{B}$	$A \vee \overline{B}$	$A \vee \overline{B} + 1$
4	0	0	1	1	$-1$	$-1$	$0$
5	0	1	0	0	$AB$	$A\overline{B} + A$	$A\overline{B} + A + 1$
6	0	1	0	1	$B$	$A\overline{B} + (A \vee B)$	$A\overline{B} + (A \vee B) + 1$
7	0	1	1	0	$A \oplus \overline{B}$	$A - B - 1$	$A - B$
8	0	1	1	1	$\overline{A} \vee B$	$A\overline{B} - 1$	$A\overline{B}$
9	1	0	0	0	$A\overline{B}$	$AB + A$	$AB + A + 1$
10	1	0	0	1	$A \oplus B$	$A + B$	$A + B + 1$
11	1	0	1	0	$\overline{B}$	$AB + (A \vee \overline{B})$	$AB + (A \vee \overline{B}) + 1$
12	1	0	1	1	$\overline{AB}$	$\overline{AB} - 1$	$\overline{AB}$
13	1	1	0	0	$0$	$A + A$	$A + A + 1$
14	1	1	0	1	$\overline{AB}$	$A + (A \vee B)$	$A + (A \vee B) + 1$
15	1	1	1	0	$\overline{A \vee \overline{B}}$	$A + (A \vee \overline{B})$	$A + (A \vee \overline{B}) + 1$
16	1	1	1	1	$\overline{A}$	$A - 1$	$A$

# Ускоренный перенос

$$P_i = D_i \vee F_i P_{i-1}.$$

$$\begin{cases} P_1 = D_1 + P_0 F_1 \\ P_2 = D_2 + P_1 F_2 \\ P_3 = D_3 + P_2 F_3 \\ P_4 = D_4 + P_3 F_4 \end{cases}$$