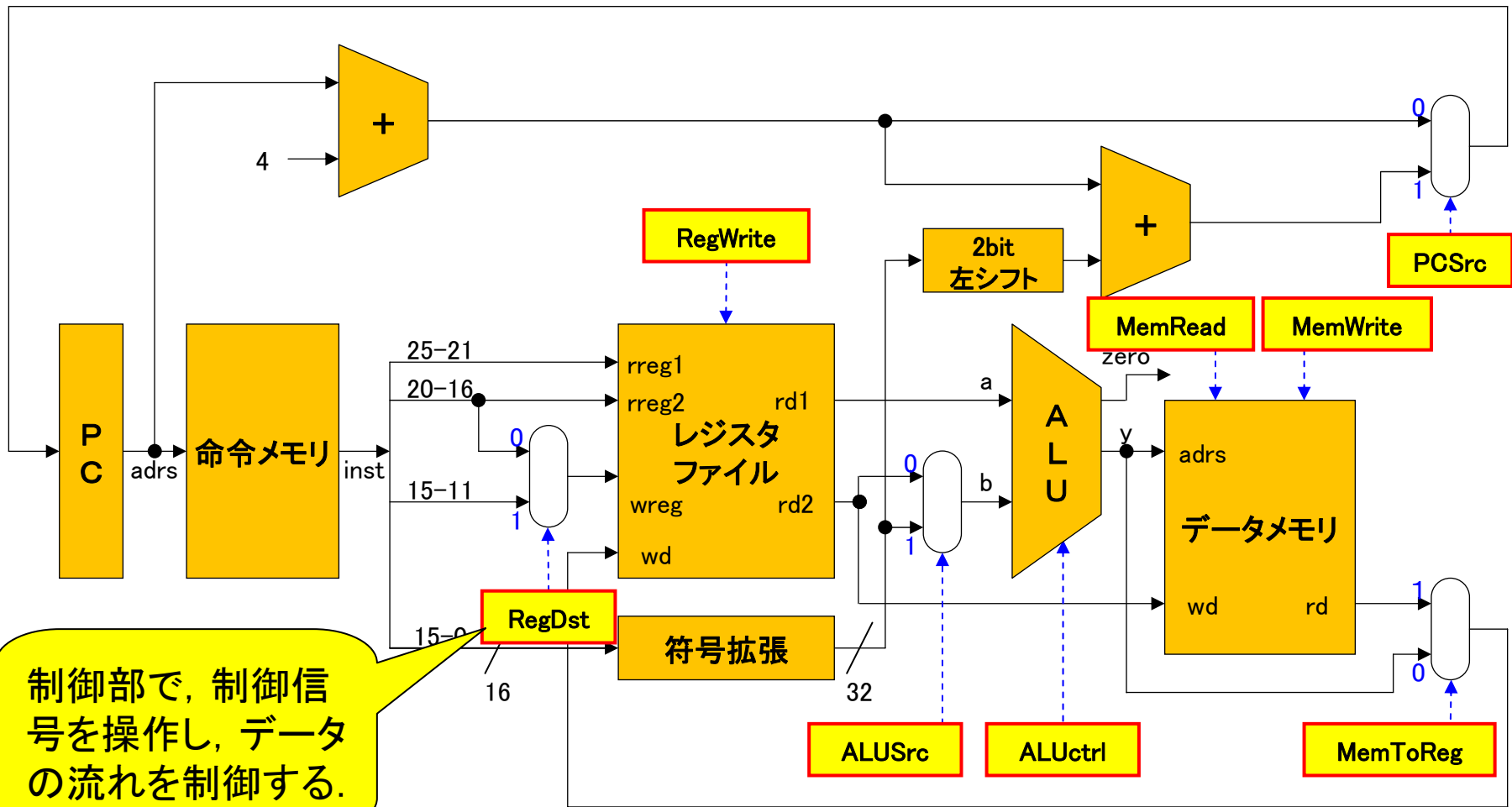


# データパスの構築Ⅱ ～制御部の構築～

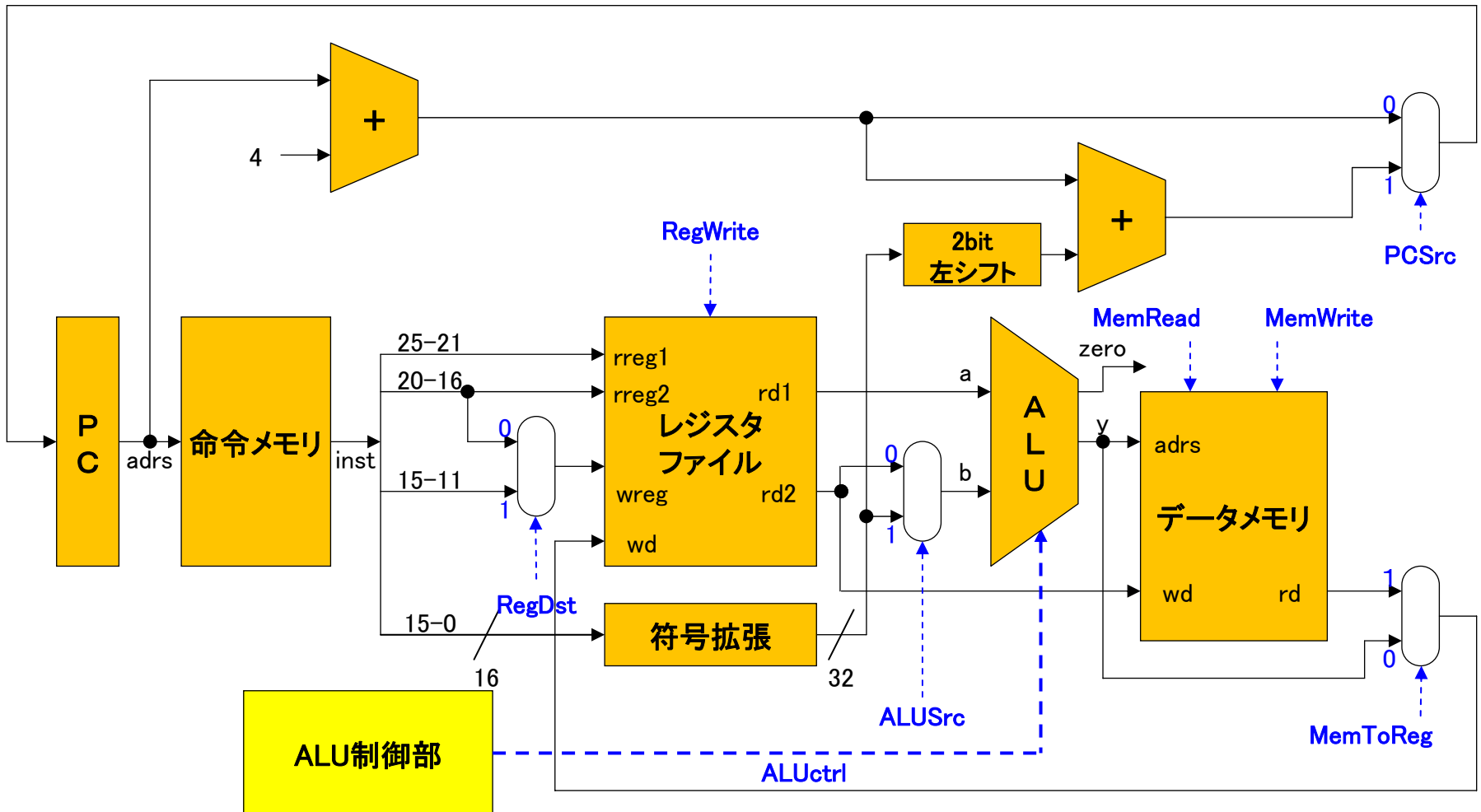
(教科書5.3節)

# 制御部の設計



制御部で、制御信号を操作し、データの流れを制御する。

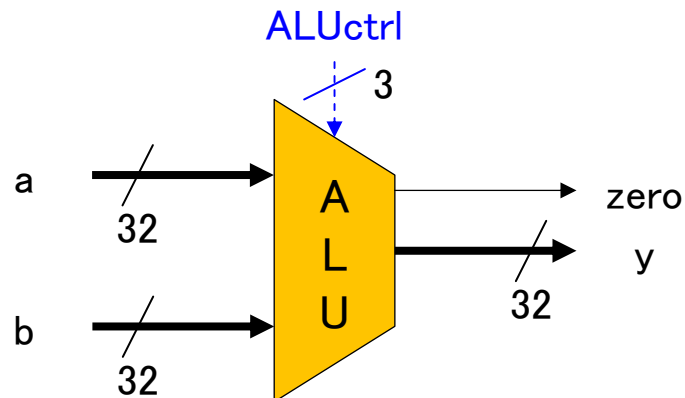
# ALU制御部の設計(1)



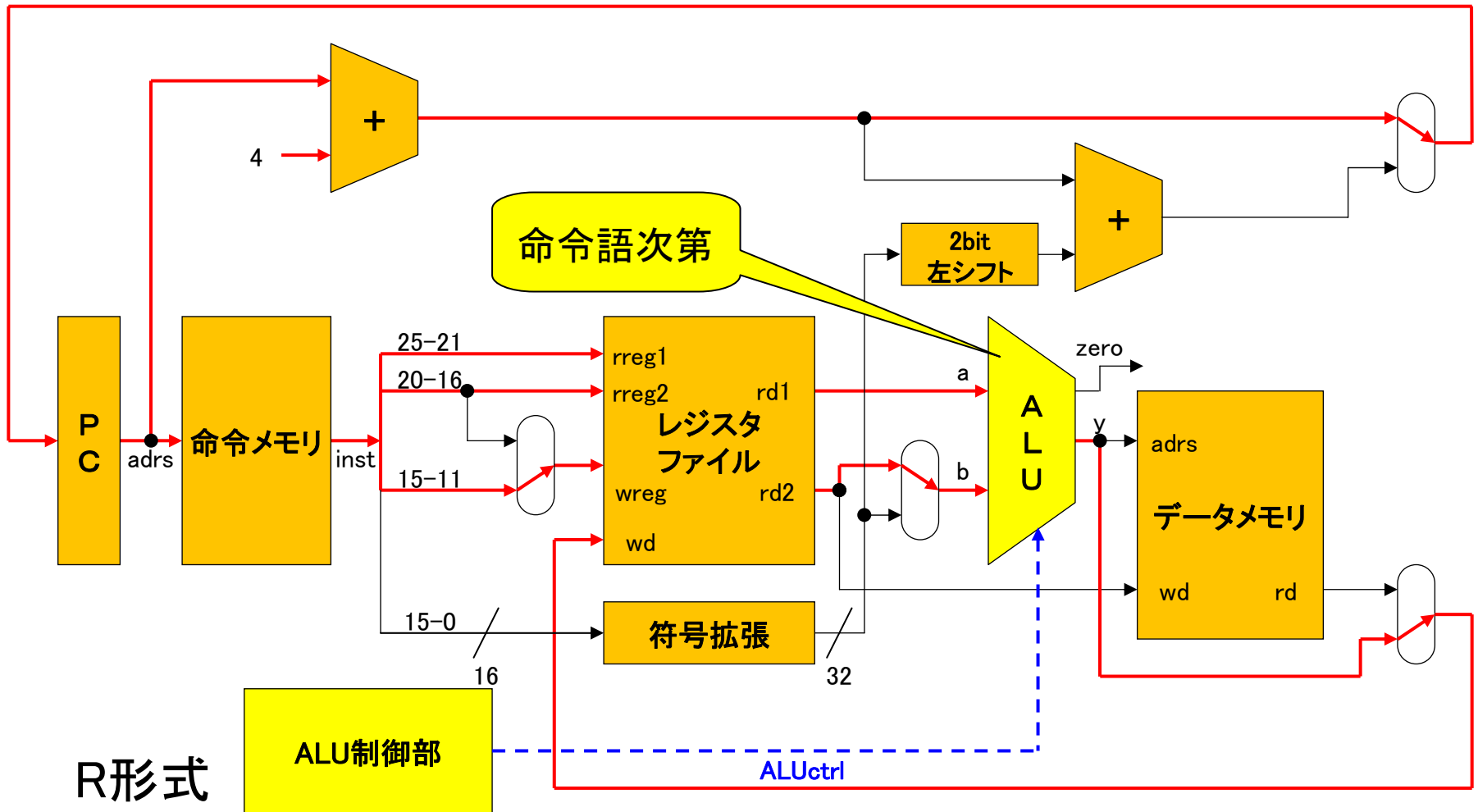
# ALU制御部の設計(2)

ALU入出力仕様:

信号の意味	入出力	信号名	ビット幅
演算パラメータ1	入力	a	32
演算パラメータ2	入力	b	32
演算結果	出力	y	32
ゼロ判定(結果が0のときに1)	出力	zero	1
ALU制御(000: AND, 001: OR, 010: +, 110: -, 111: slt)	入力	ALUctrl	3



# ALU制御部の設計(3)

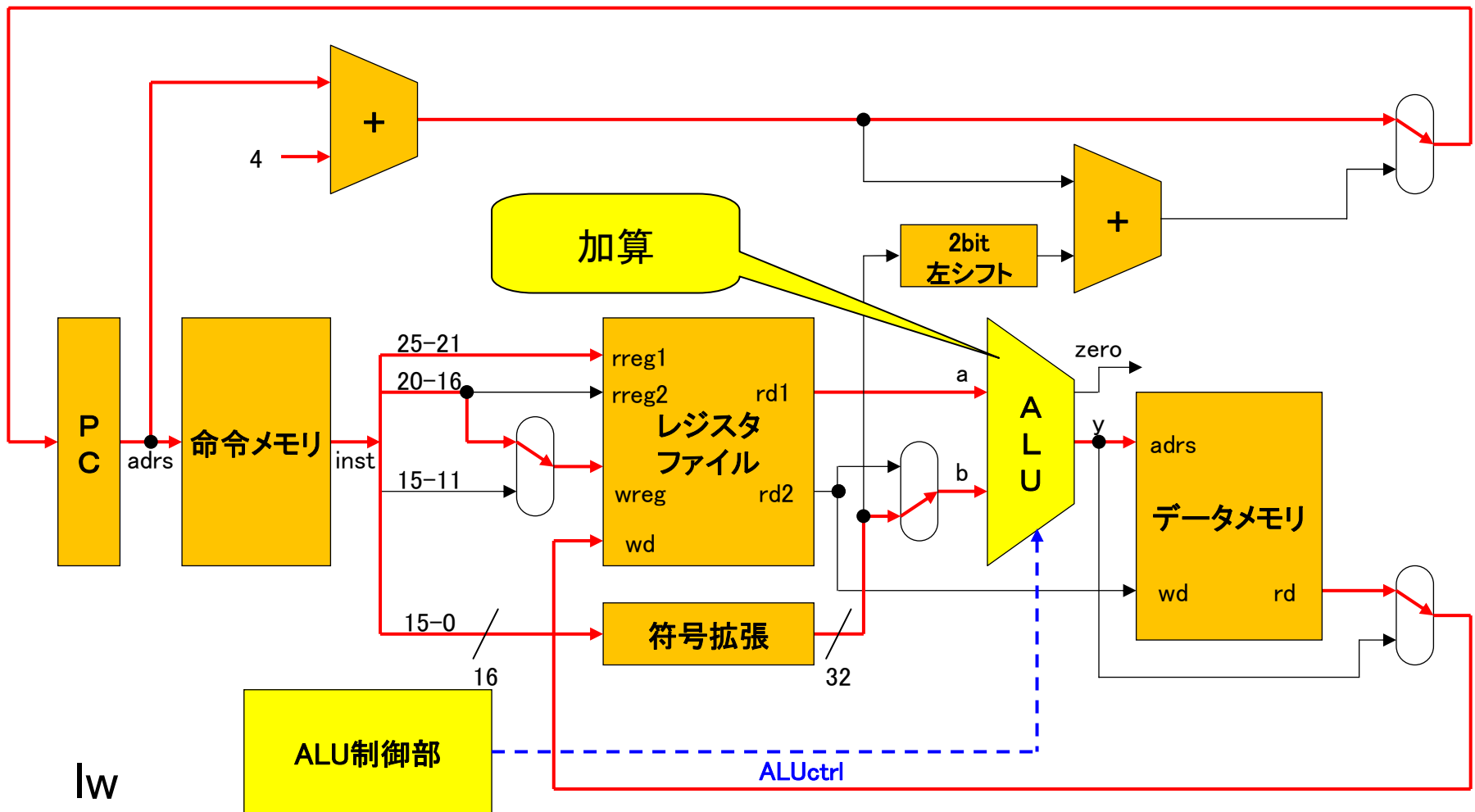


R形式

ALU制御部

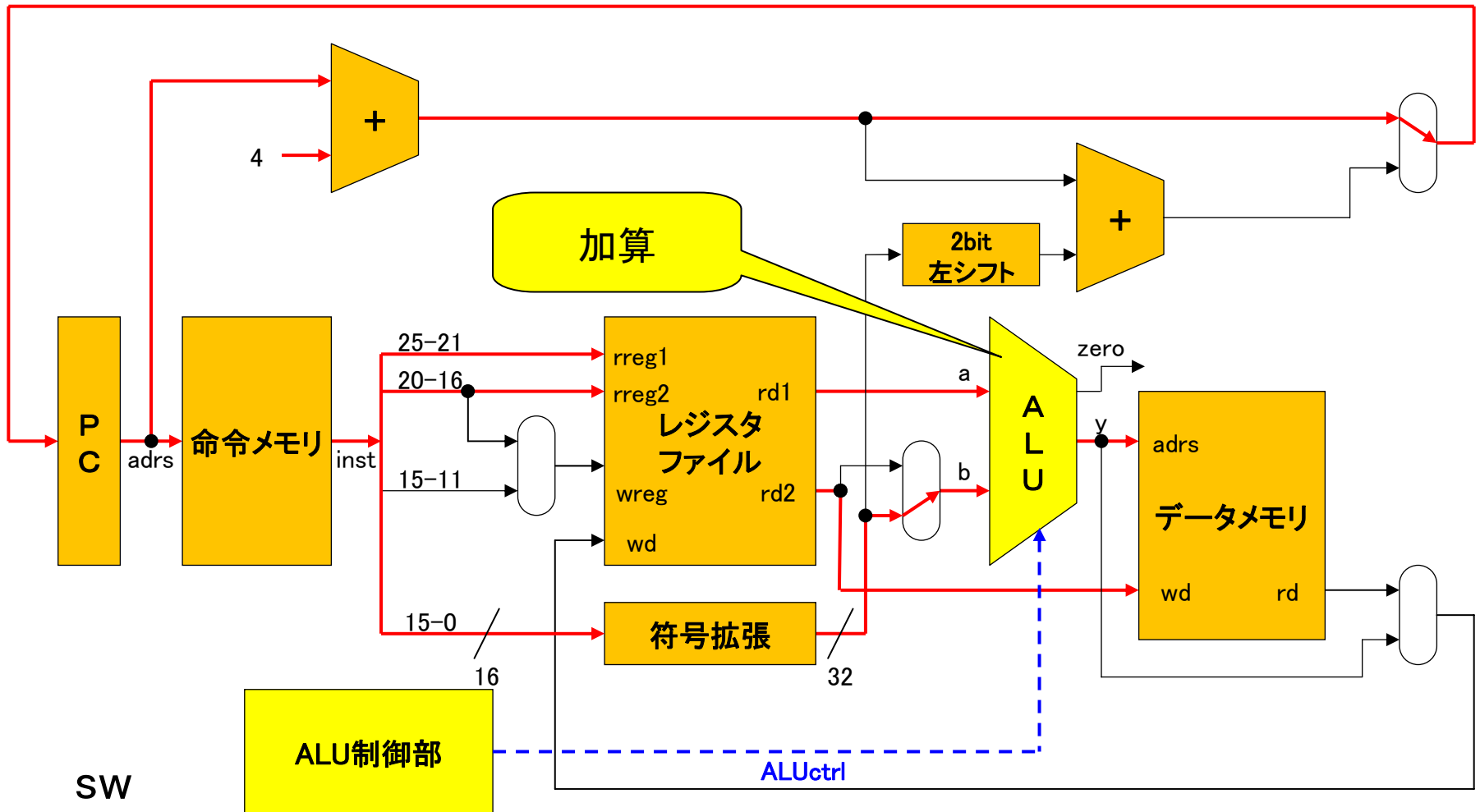
ALUctrl

# ALU制御部の設計(4)



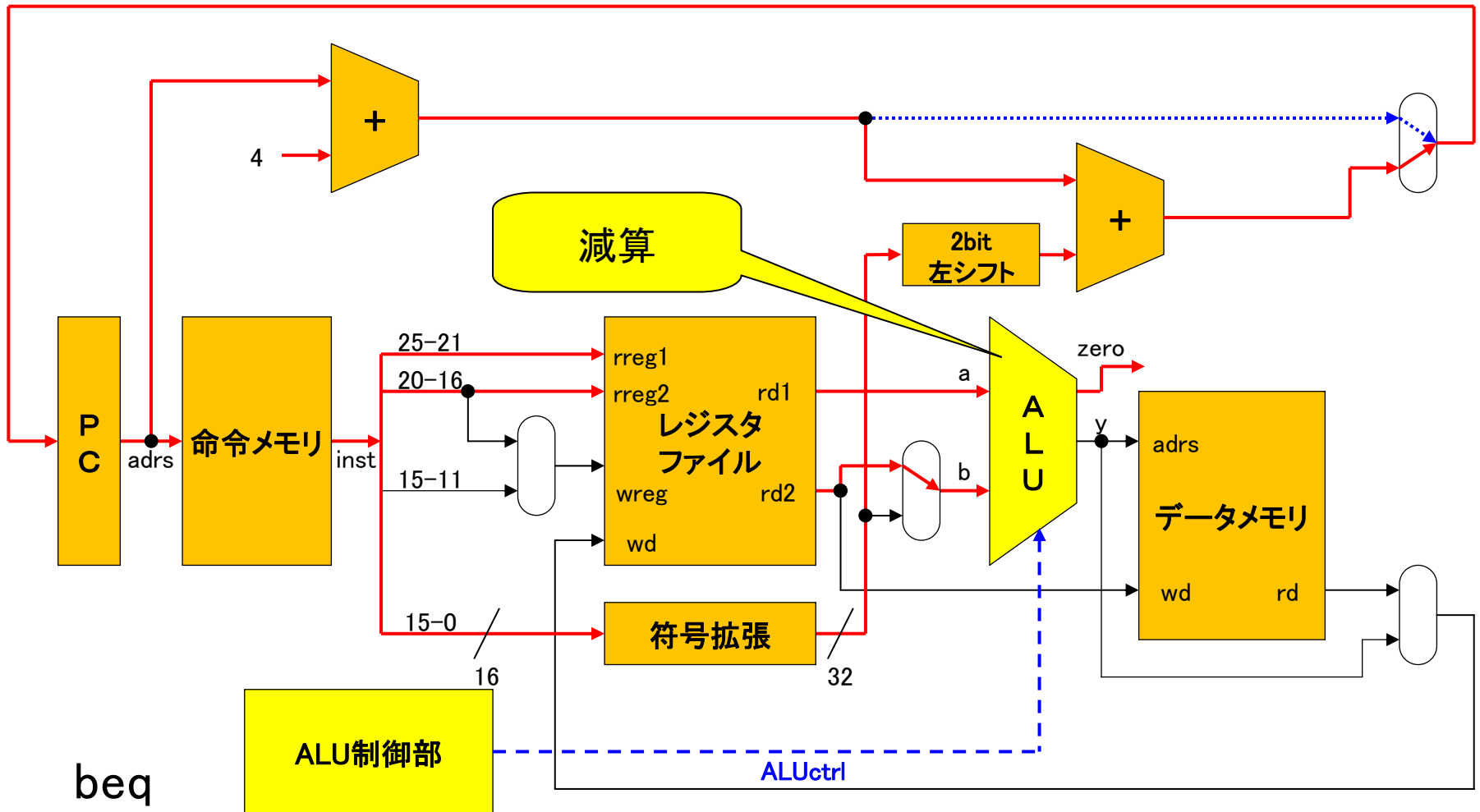
lw

# ALU制御部の設計(5)



SW

# ALU制御部の設計(6)

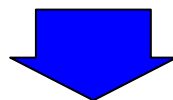




# ALU制御部の設計(7)

ALUの制御パターンは3パターン:

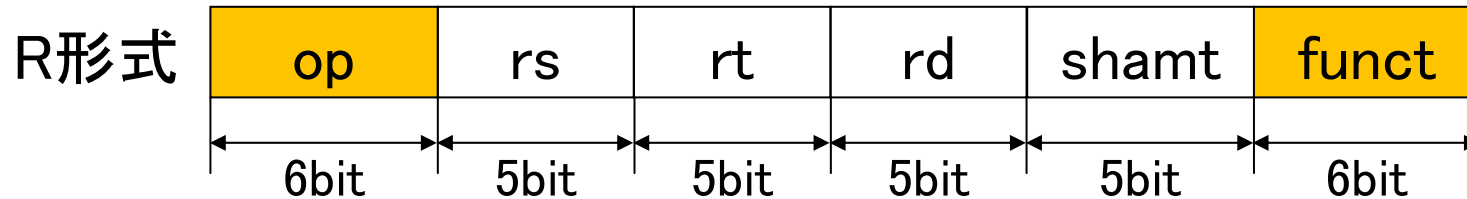
- 命令語次第(R形式)
- 加算(lw, sw命令)
- 減算(beq命令)



どのパターンの制御をするか指定する入力信号(ALUOp, 2bit)が必要.

信号値	意味
00	加算(lw, sw命令)
01	減算(beq命令)
1X	命令語の指定どおりの演算(R形式)

# ALU制御部の設計(8)



命令	op	funct	ALUctrl
add \$O, \$Δ, \$□	000000	100000	010
sub \$O, \$Δ, \$□	000000	100010	110
slt \$O, \$Δ, \$□	000000	101010	111
and \$O, \$Δ, \$□	000000	100100	000
or \$O, \$Δ, \$□	000000	100101	001

# ALU制御部の設計(9)

ALU制御部の仕様:

信号の意味	入出力	信号名	ビット幅
ALU制御パターン	入力	ALUOp	2
R形式命令語の funct	入力	funct	6
ALU制御	出力	ALUctrl	3

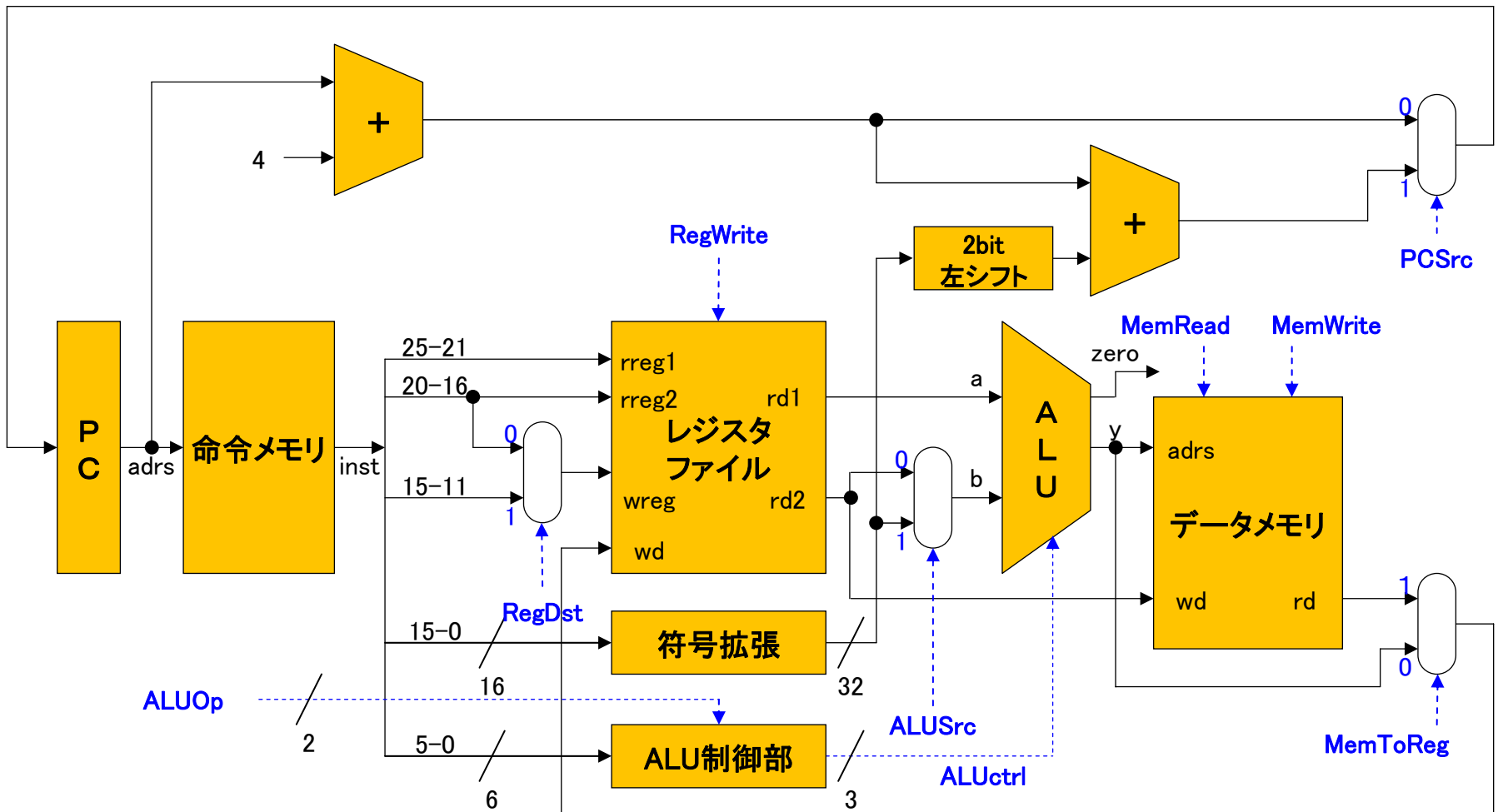


# ALU制御部の設計(10)

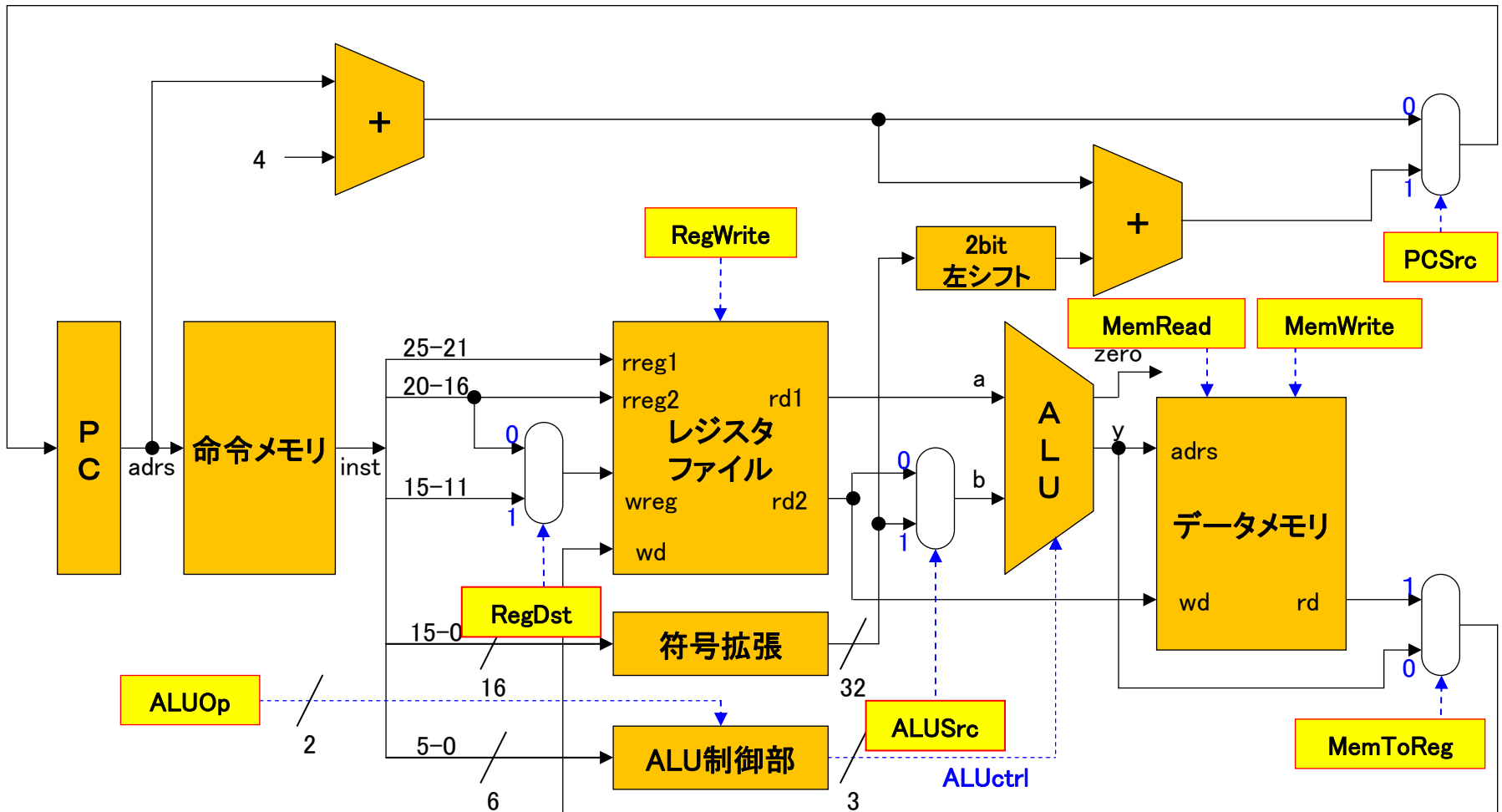
ALU制御部の仕様:

命令	ALUOp	funct	ALUctrl
lw \$O, n(\$Δ)	00	XXXXXX	010
sw \$O, n(\$Δ)	00	XXXXXX	010
beq \$O, \$Δ, Label	01	XXXXXX	110
add \$O, \$Δ, \$□	1X	XX0000	010
sub \$O, \$Δ, \$□	1X	XX0010	110
slt \$O, \$Δ, \$□	1X	XX1010	111
and \$O, \$Δ, \$□	1X	XX0100	000
or \$O, \$Δ, \$□	1X	XX0101	001

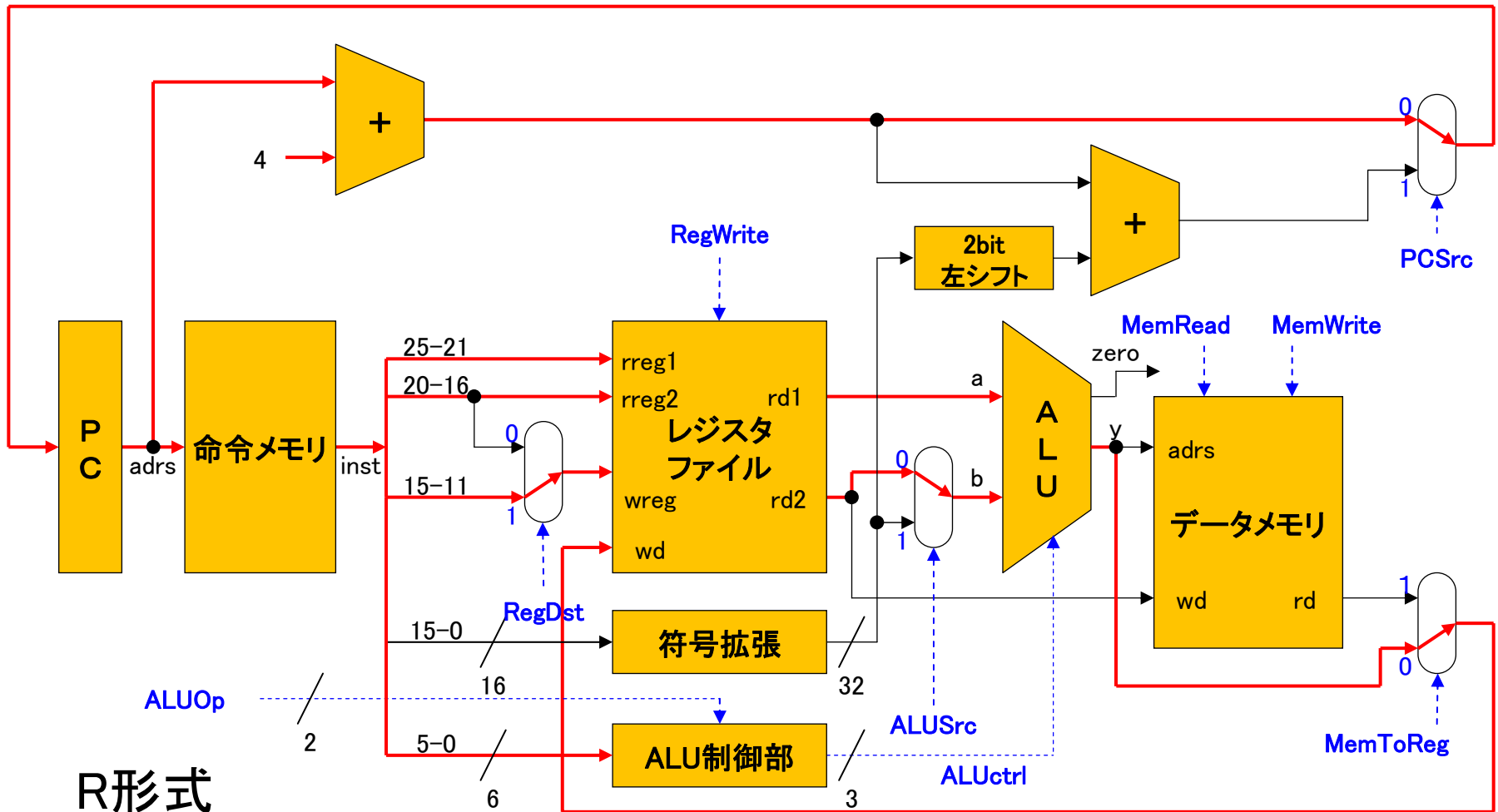
# ALU制御部の設計(11)



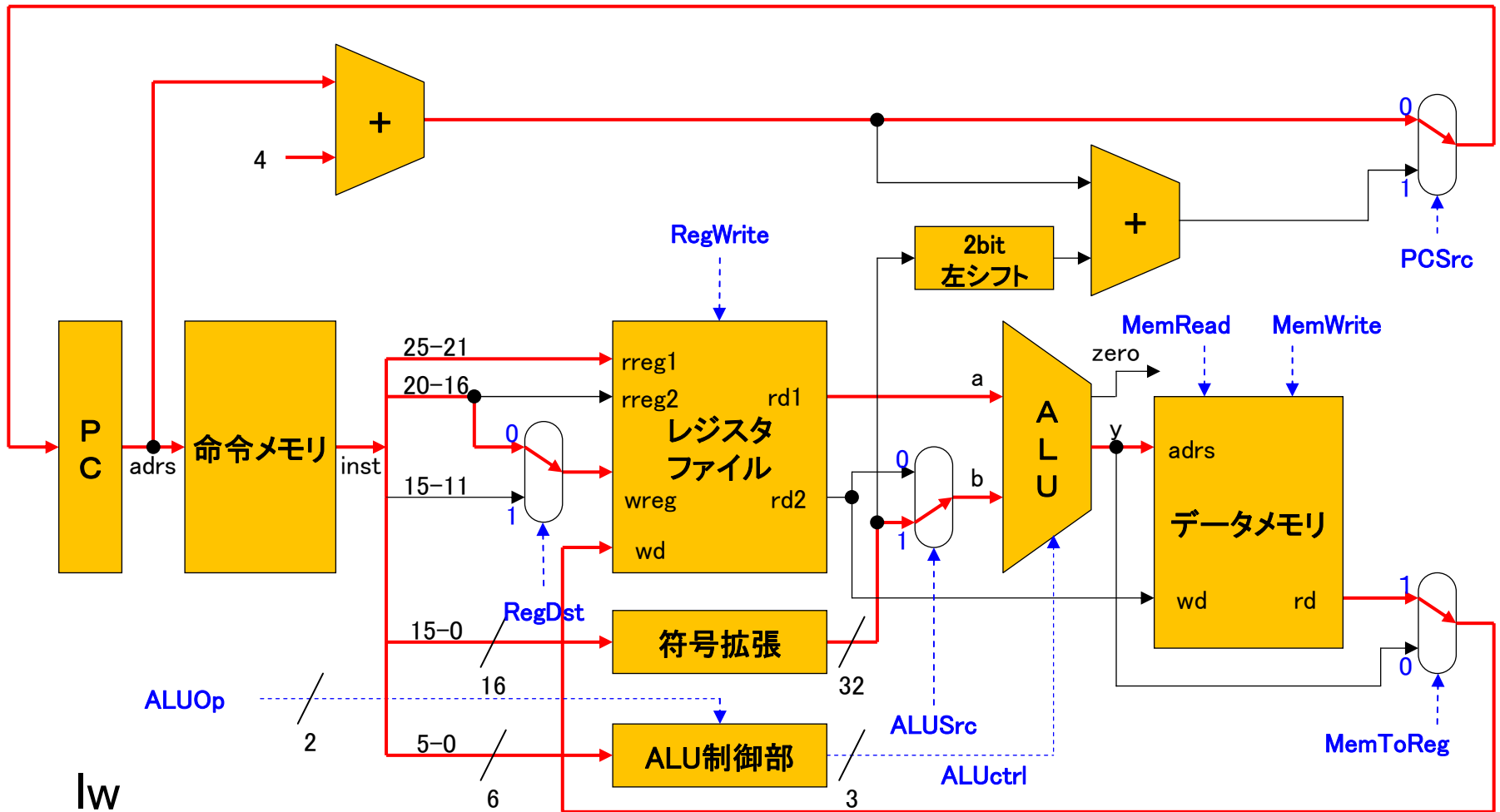
# 主制御部の設計(1)



# 主制御部の設計(2)



# 主制御部の設計(3)

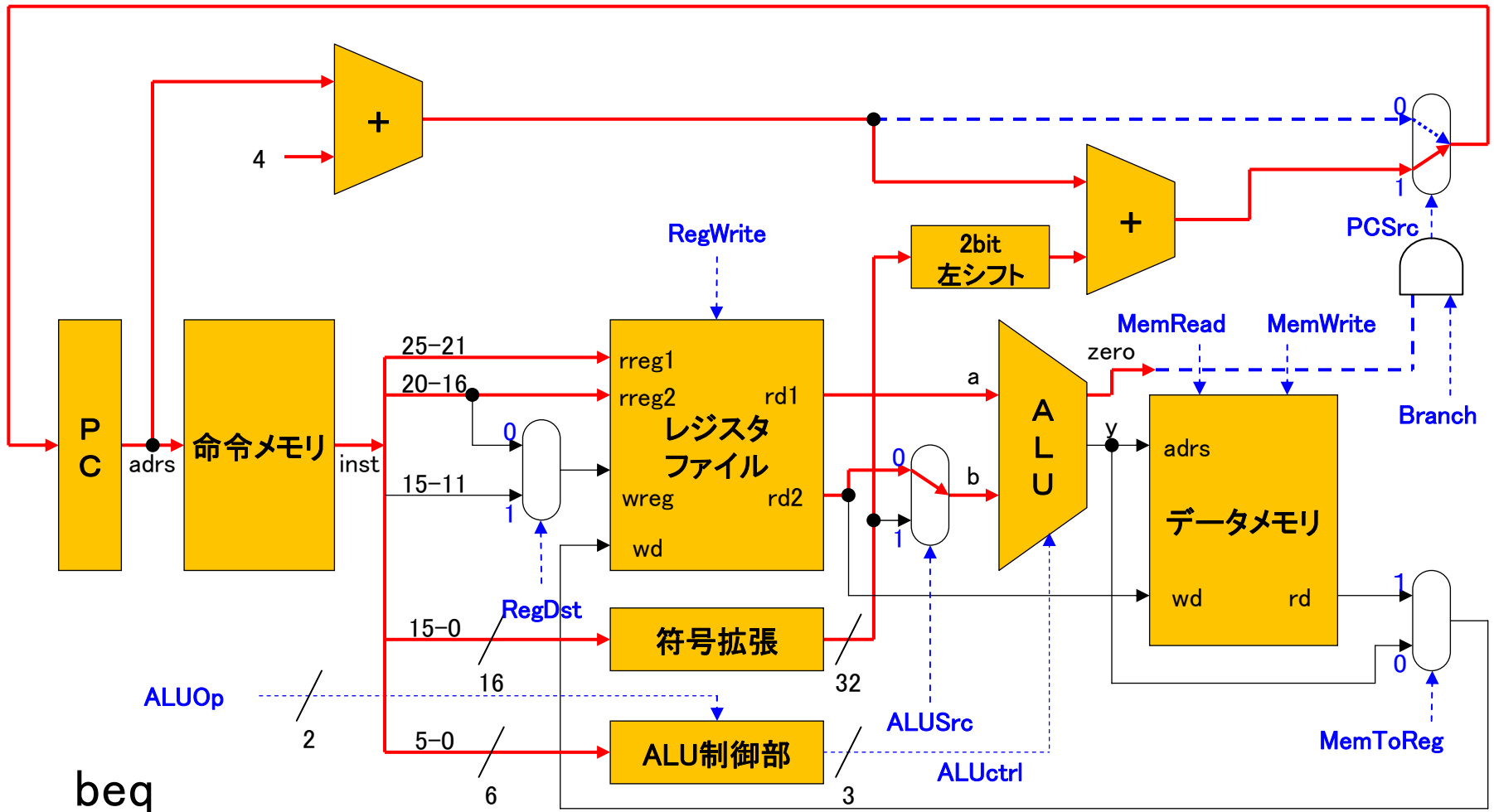


lw





# 主制御部の設計(5)

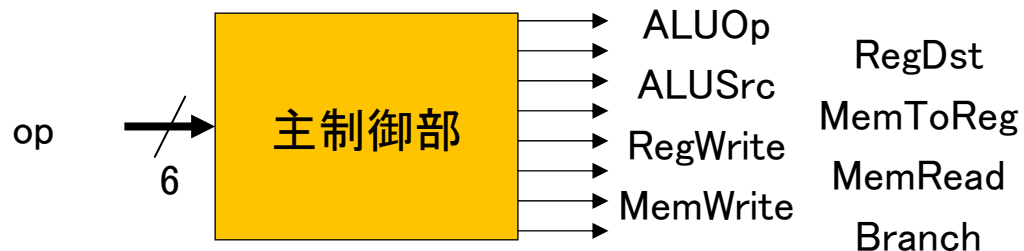


beq

# 主制御部の設計(6)

主制御部の仕様:

命令	ALUOp	RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	Branch
lw \$O, n(\$Δ)	00	0	1	1	1	1	0	0
sw \$O, n(\$Δ)	00	X	1	X	0	0	1	0
beq \$O, \$Δ, Label	01	X	0	X	0	0	0	1
R形式	1X	1	0	0	1	0	0	0



# 主制御部の設計(7)

