

マルチサイクルを用いた実現方式

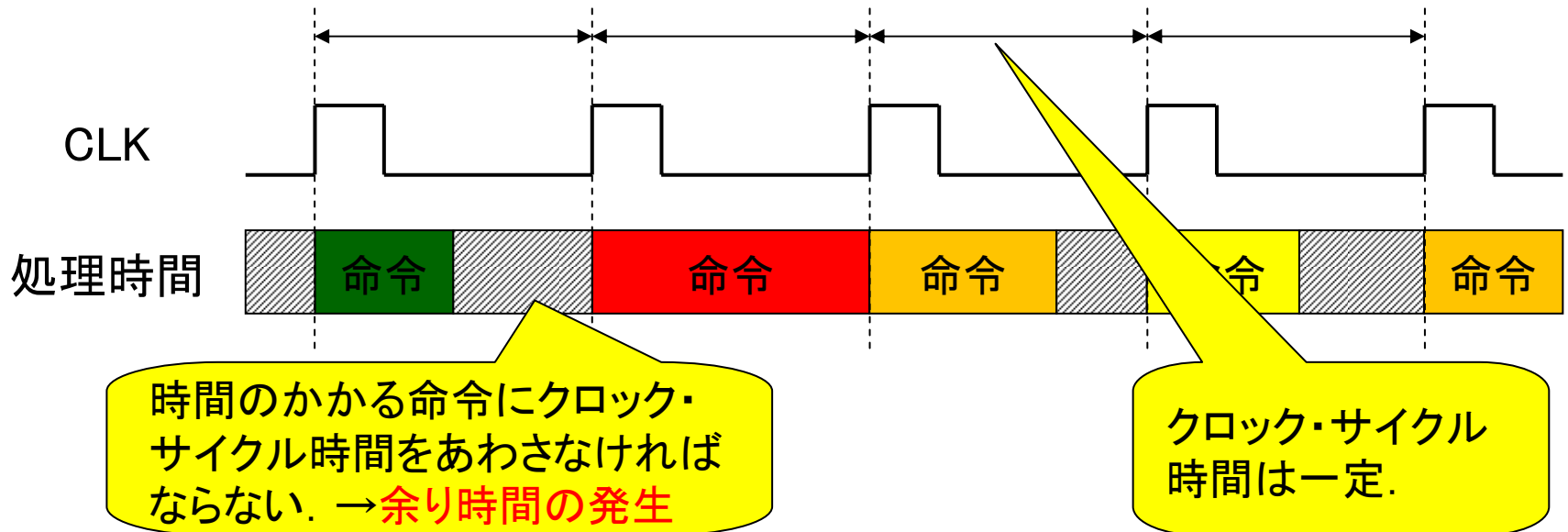
(教科書5.4節)

マルチサイクル方式(1)

2つのデータパス実現方式

- 単一クロックサイクル: 1命令を1クロックサイクルで処理
- マルチクロックサイクル: 1命令を複数クロックサイクルで処理

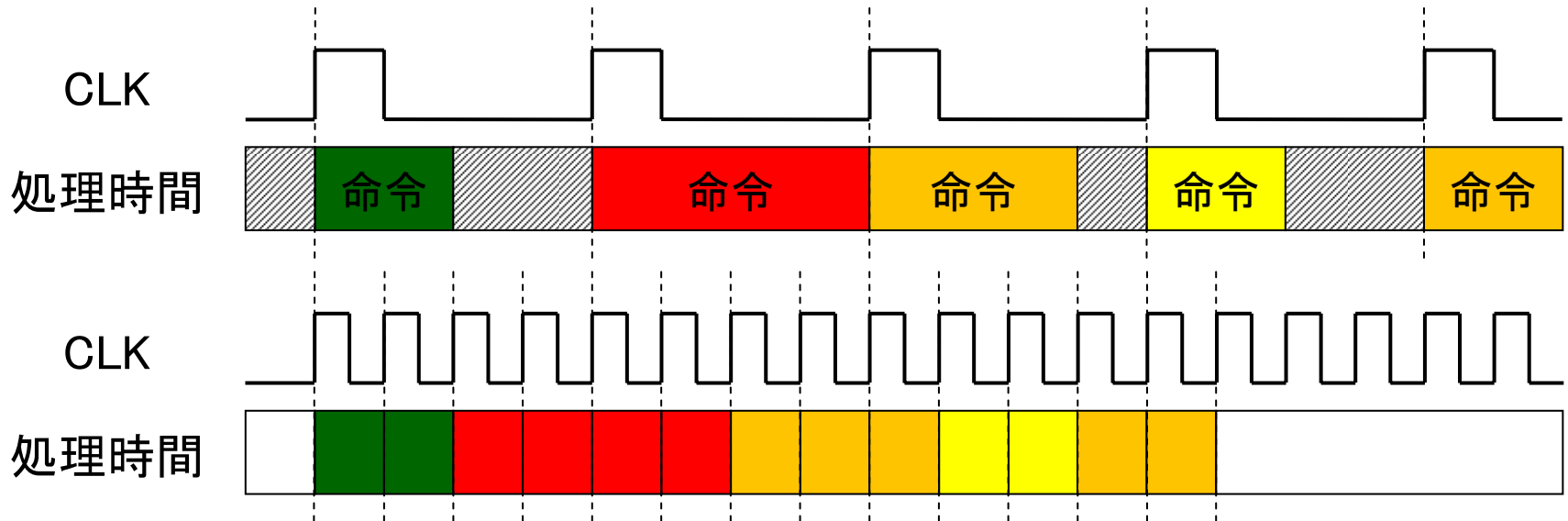
単一クロックサイクル方式は処理効率が悪い。



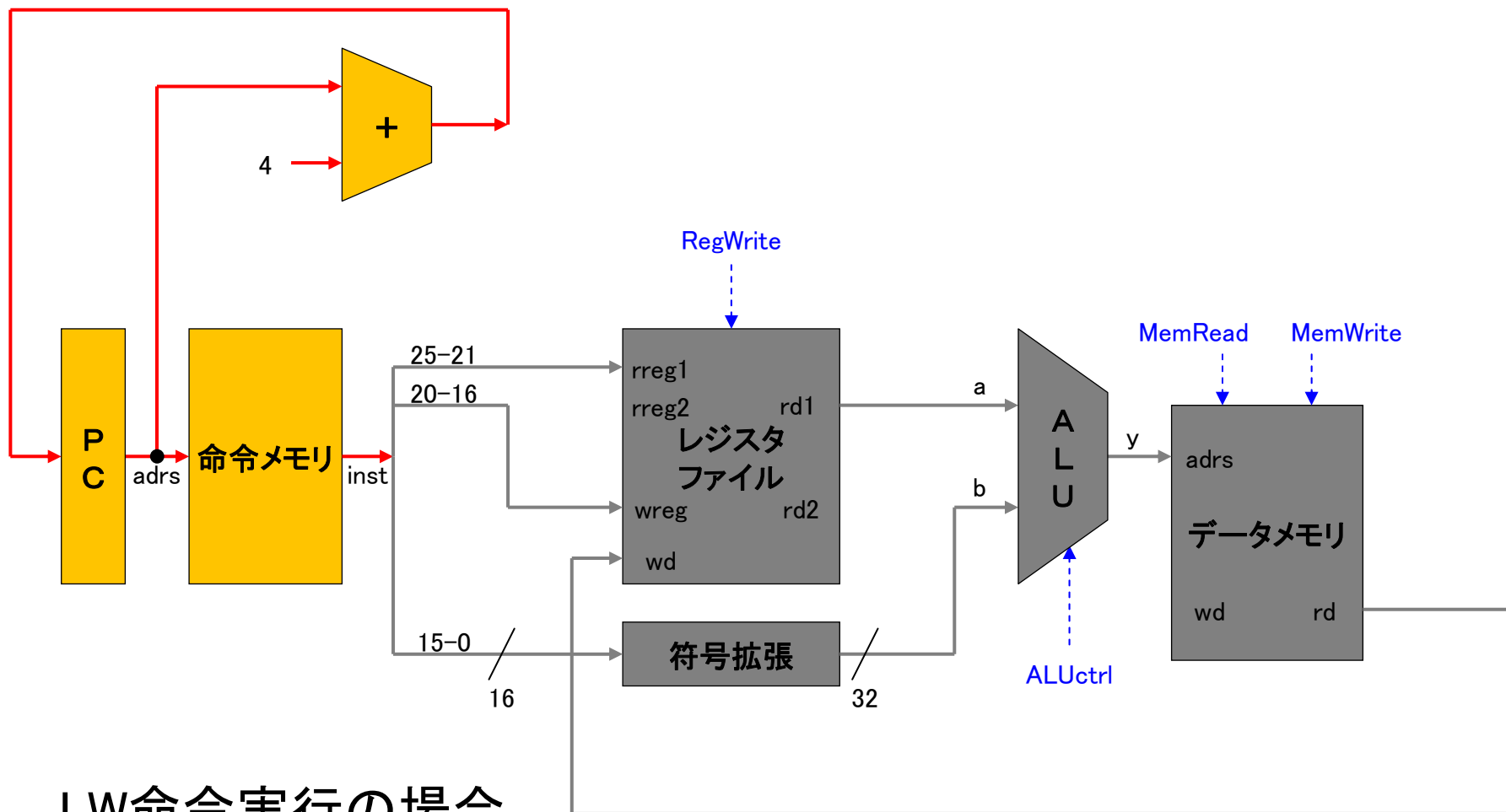
マルチサイクル方式(2)

マルチクロックサイクル方式

- 1命令の実行を(同程度の処理量の)複数小作業に分割する
 - 各小作業を1クロックサイクルで処理する
 - 1命令は複数クロックサイクルで実行される
- クロック・サイクル時間を短くする(動作周波数を高くする)

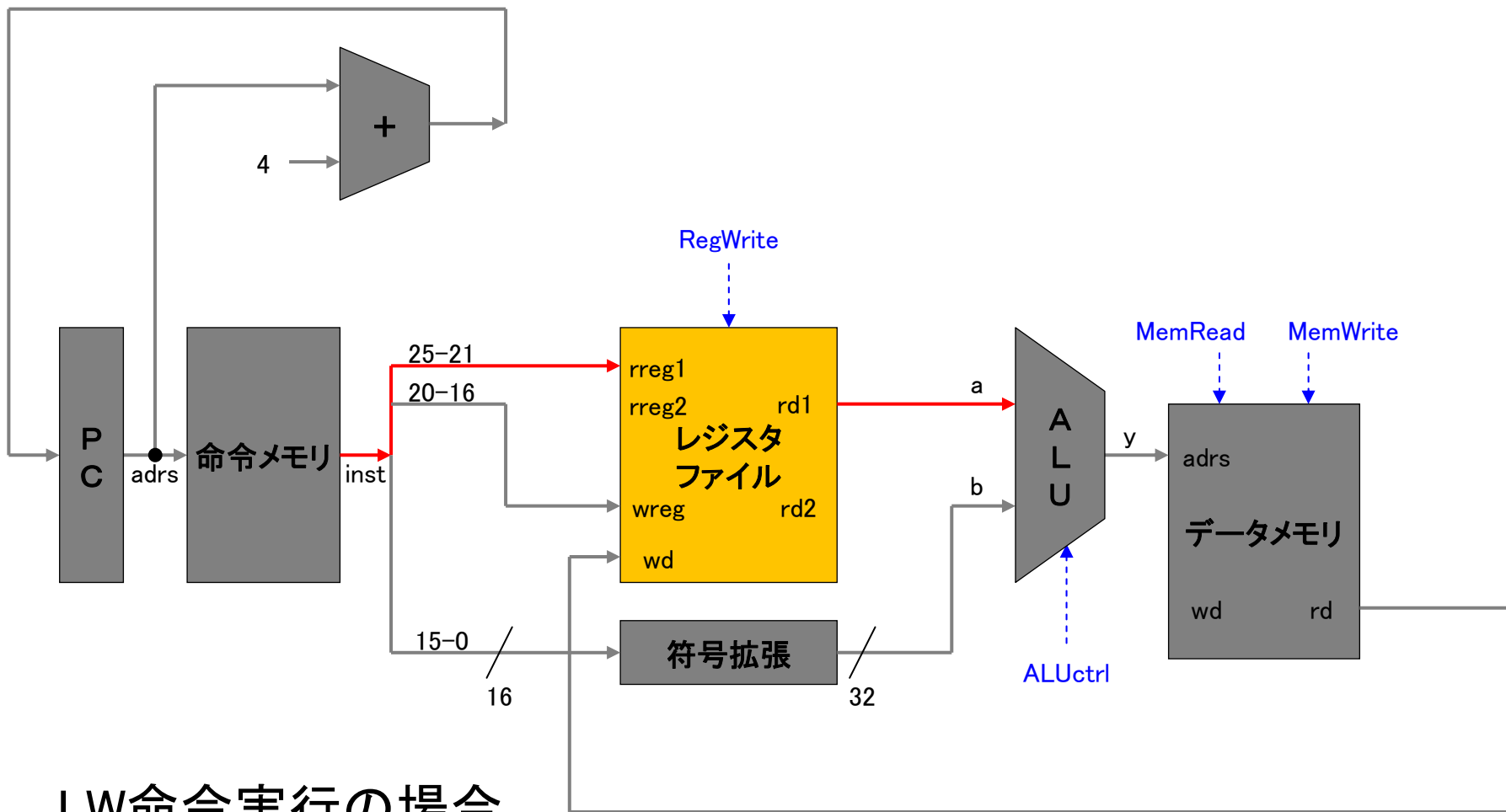


マルチサイクル処理例(1)



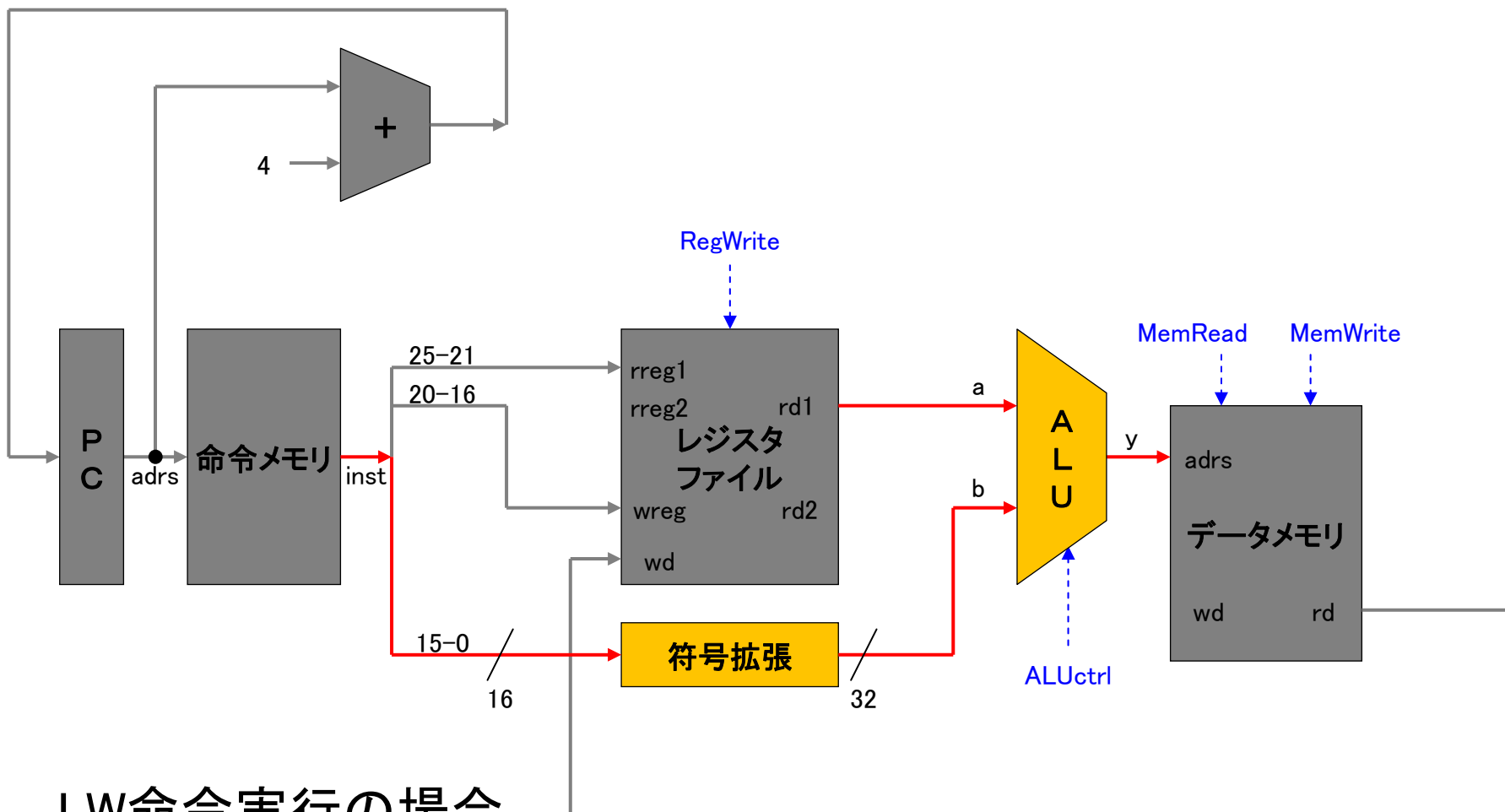
LW命令実行の場合

マルチサイクル処理例(2)



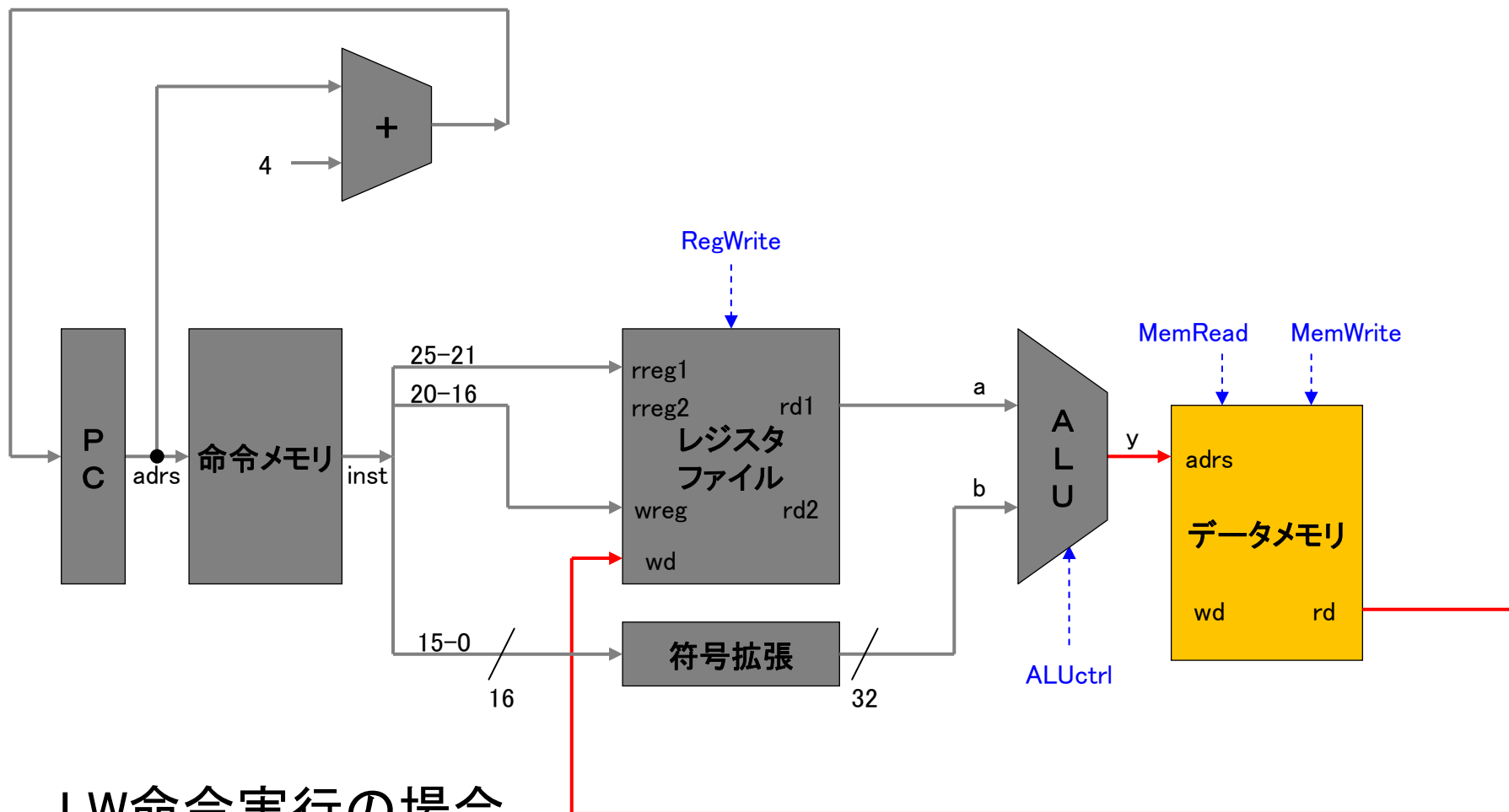
LW命令実行の場合

マルチサイクル処理例(3)



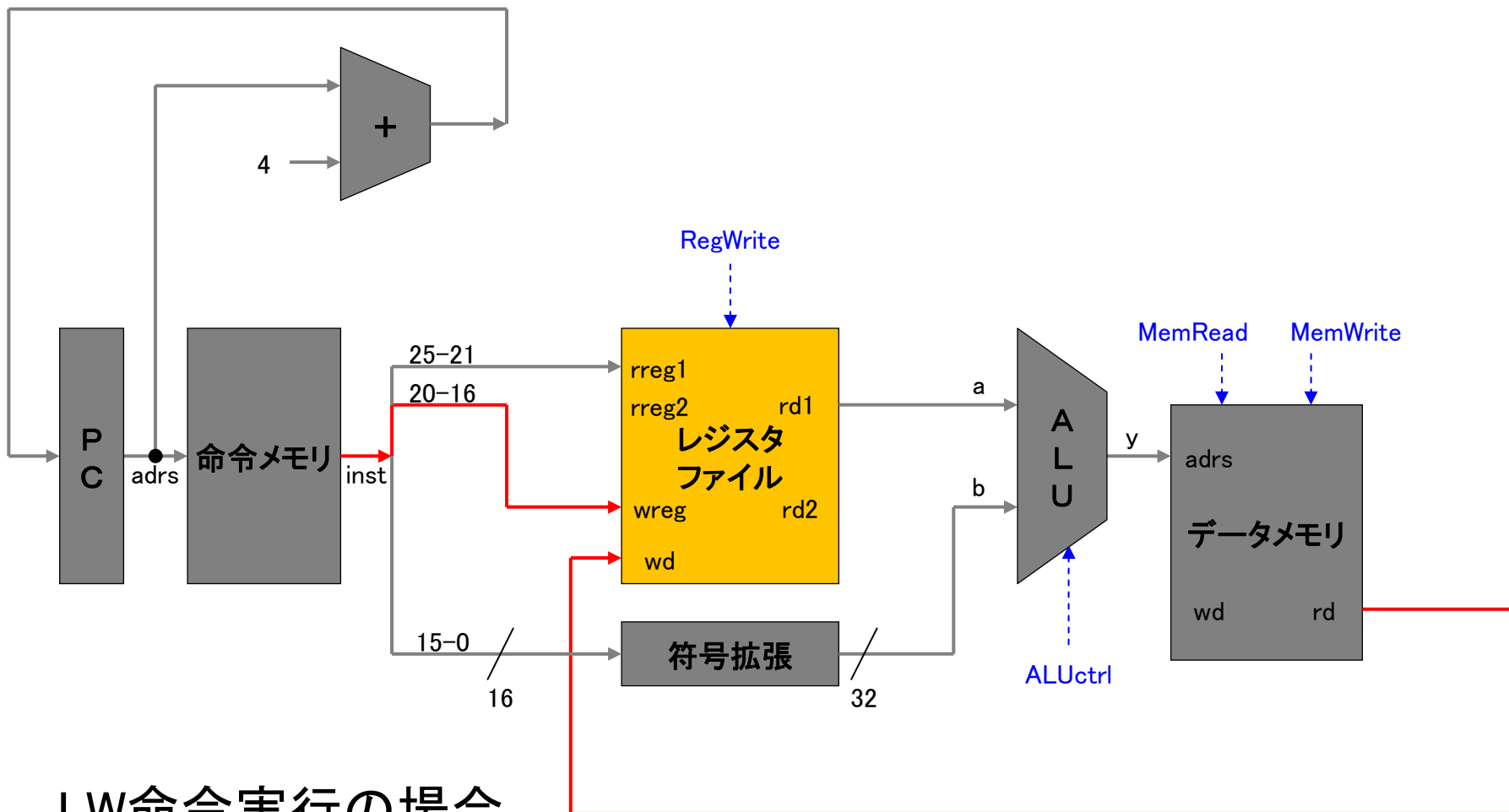
LW命令実行の場合

マルチサイクル処理例(4)



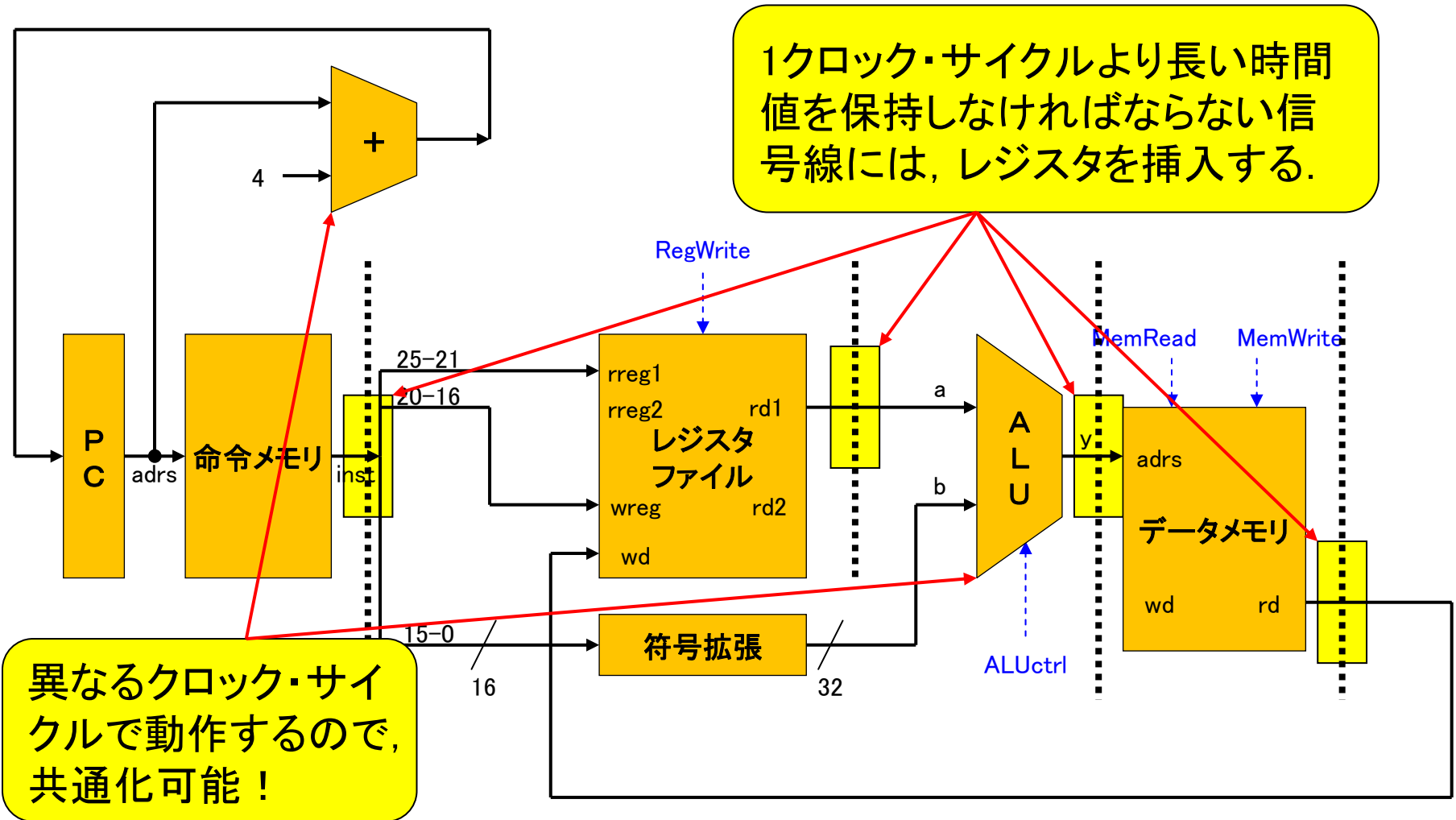
LW命令実行の場合

マルチサイクル処理例(5)

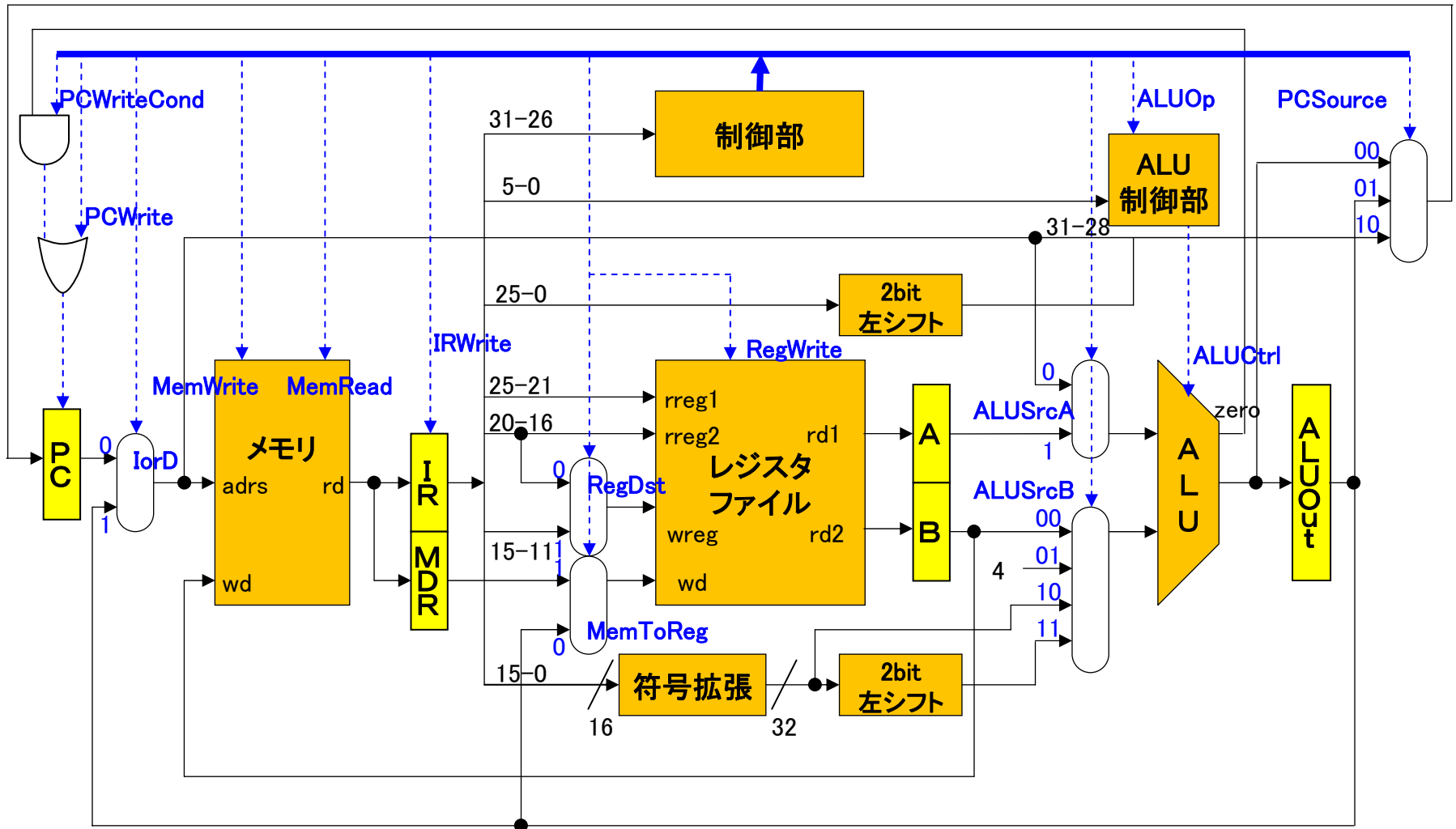


LW命令実行の場合

マルチサイクル処理例(6)



マルチサイクル・データパス(1)



マルチサイクル・データパス(2)

プログラムカウンタ(PC)

信号の意味	入出力	信号名	ビット幅
次のクロック・サイクルで更新するPCの値	入力		32
記憶してる値(現在のPCの値)	出力		32
更新要求(1で要求)	入力		1

命令レジスタ(IR): メモリから読み出した命令を記憶する.

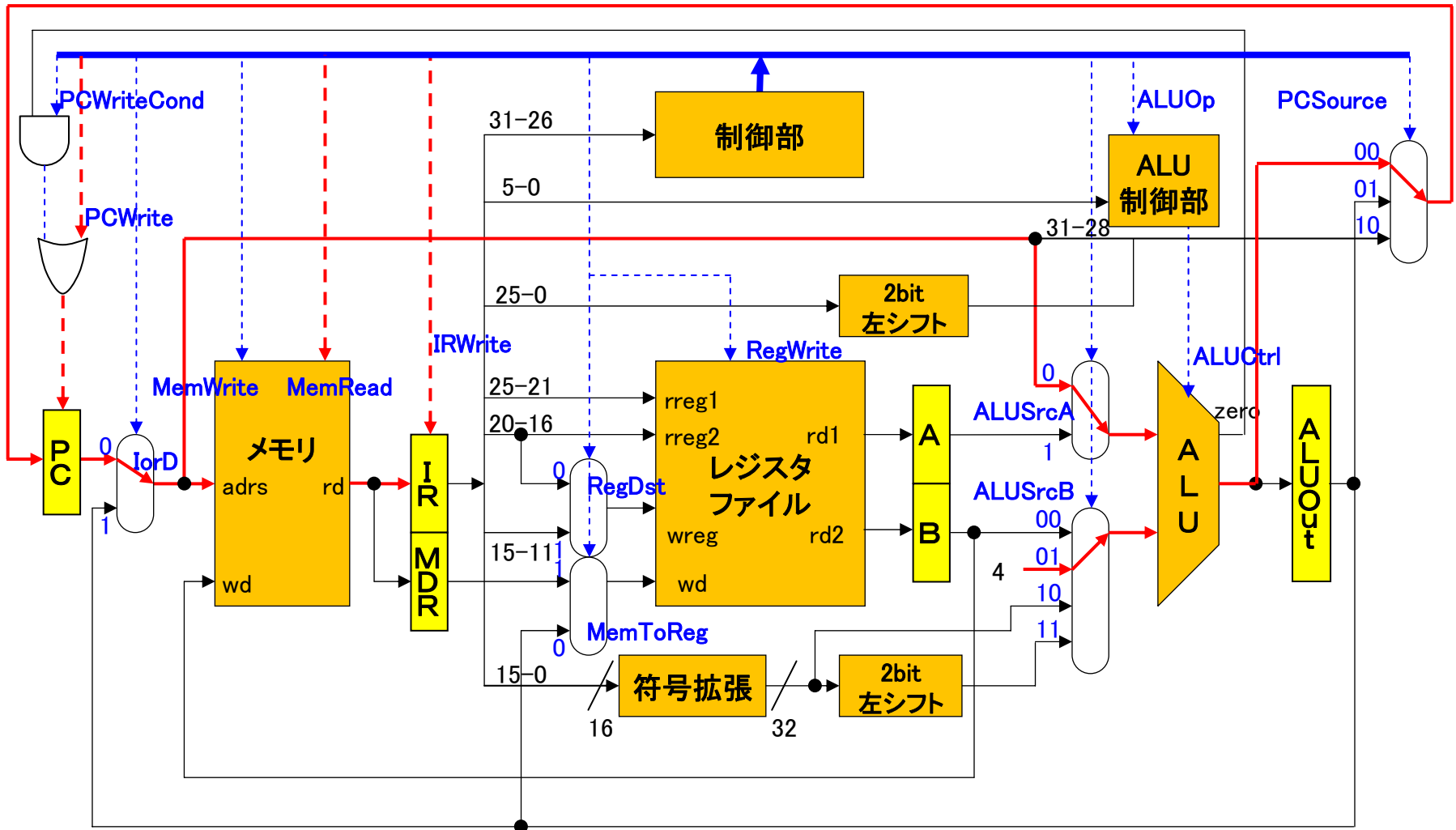
信号の意味	入出力	信号名	ビット幅
次のクロック・サイクルで記憶する命令	入力		32
記憶している命令(現在のIRの値)	出力		32
書き込み要求(1のとき要求)	入力	IRWrite	1

マルチサイクル・データパス(3)

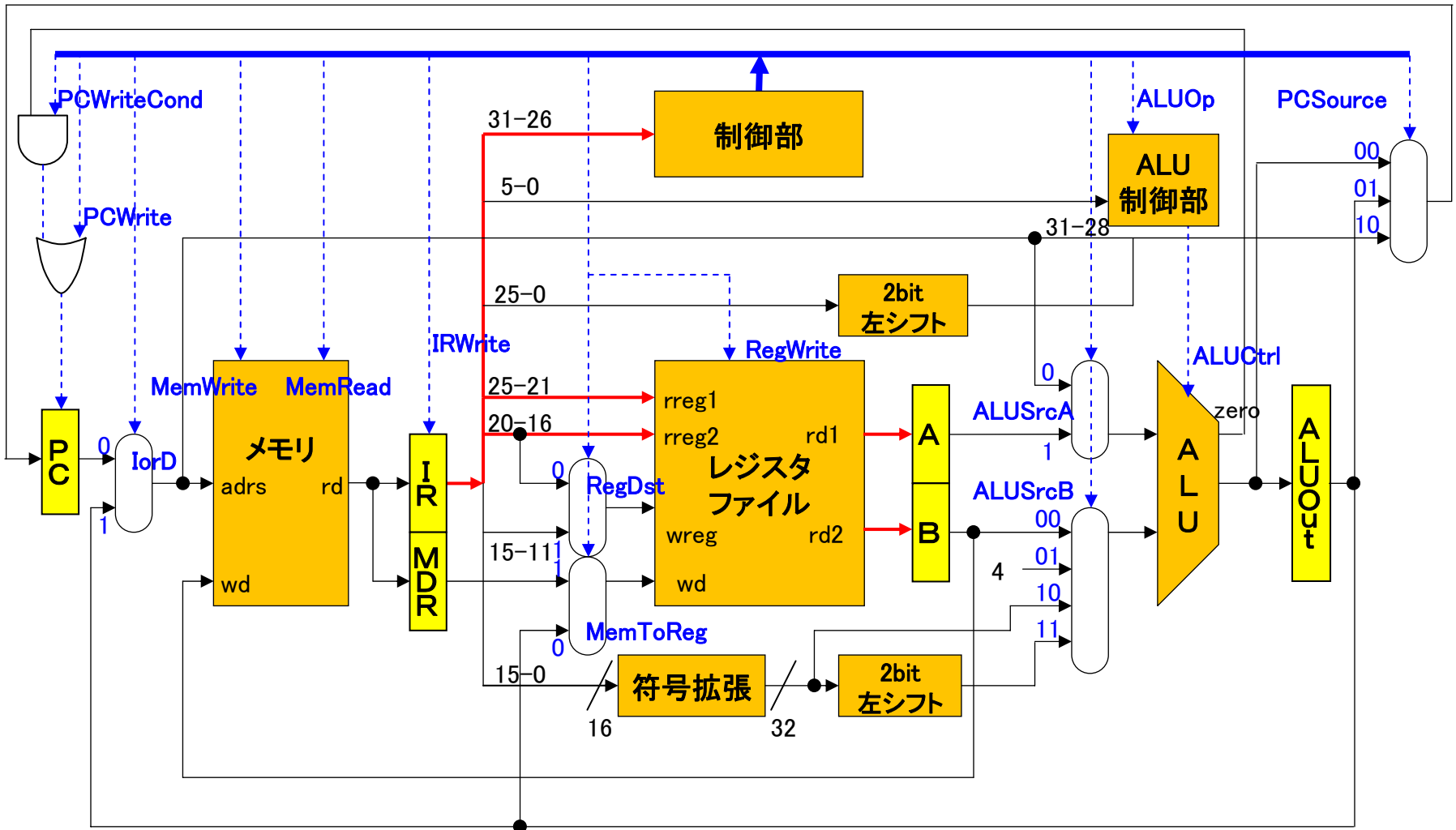
MDR(メモリデータレジスタ), A, B, ALUOut: いずれも通常のレジスタ(Dフリップフロップ)

信号の意味	入出力	信号名	ビット幅
次のクロック・サイクルで記憶するデータ	入力		32
現在記憶しているデータ	出力		32

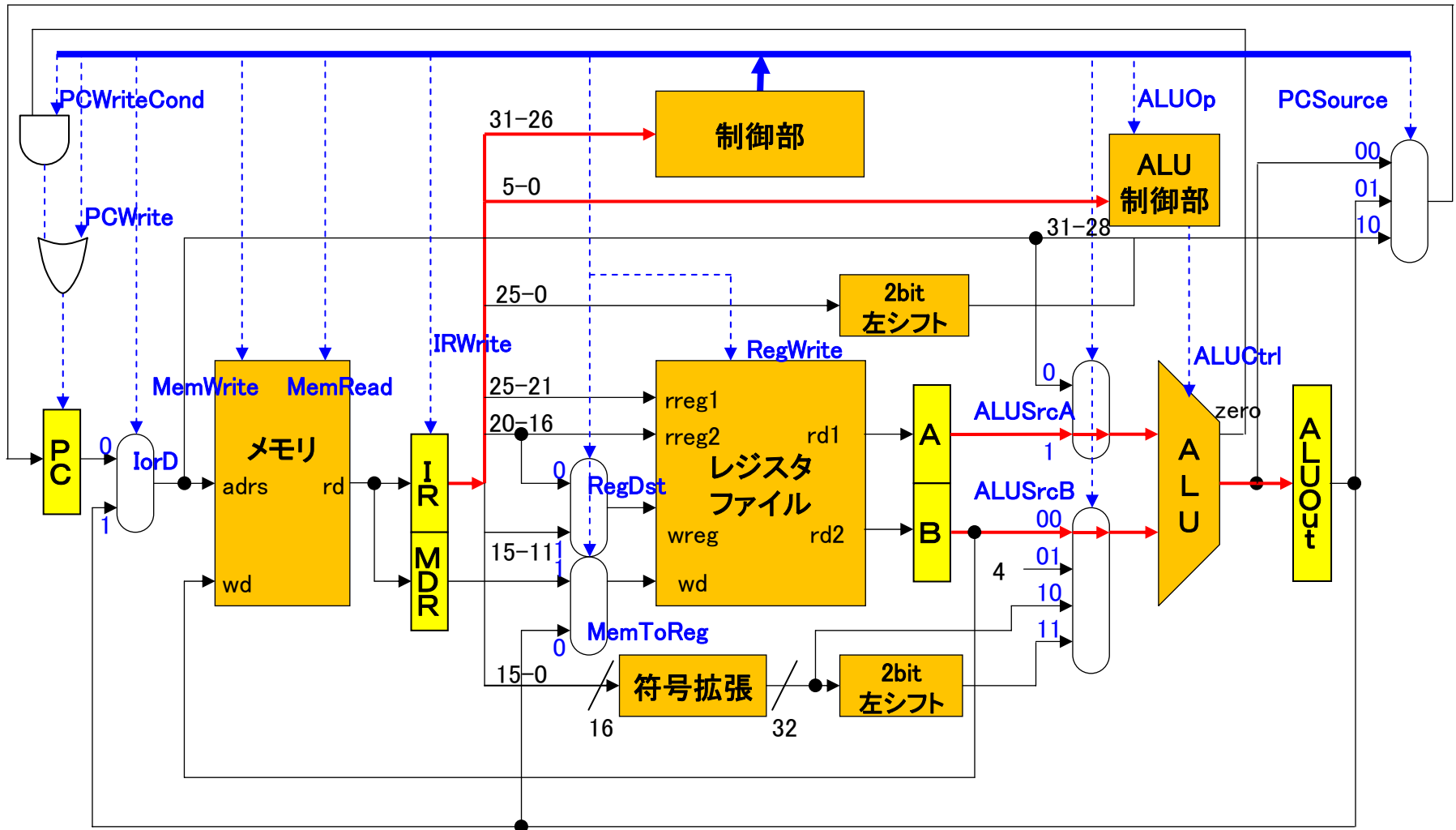
R形式命令の実行(命令取得とPC更新)



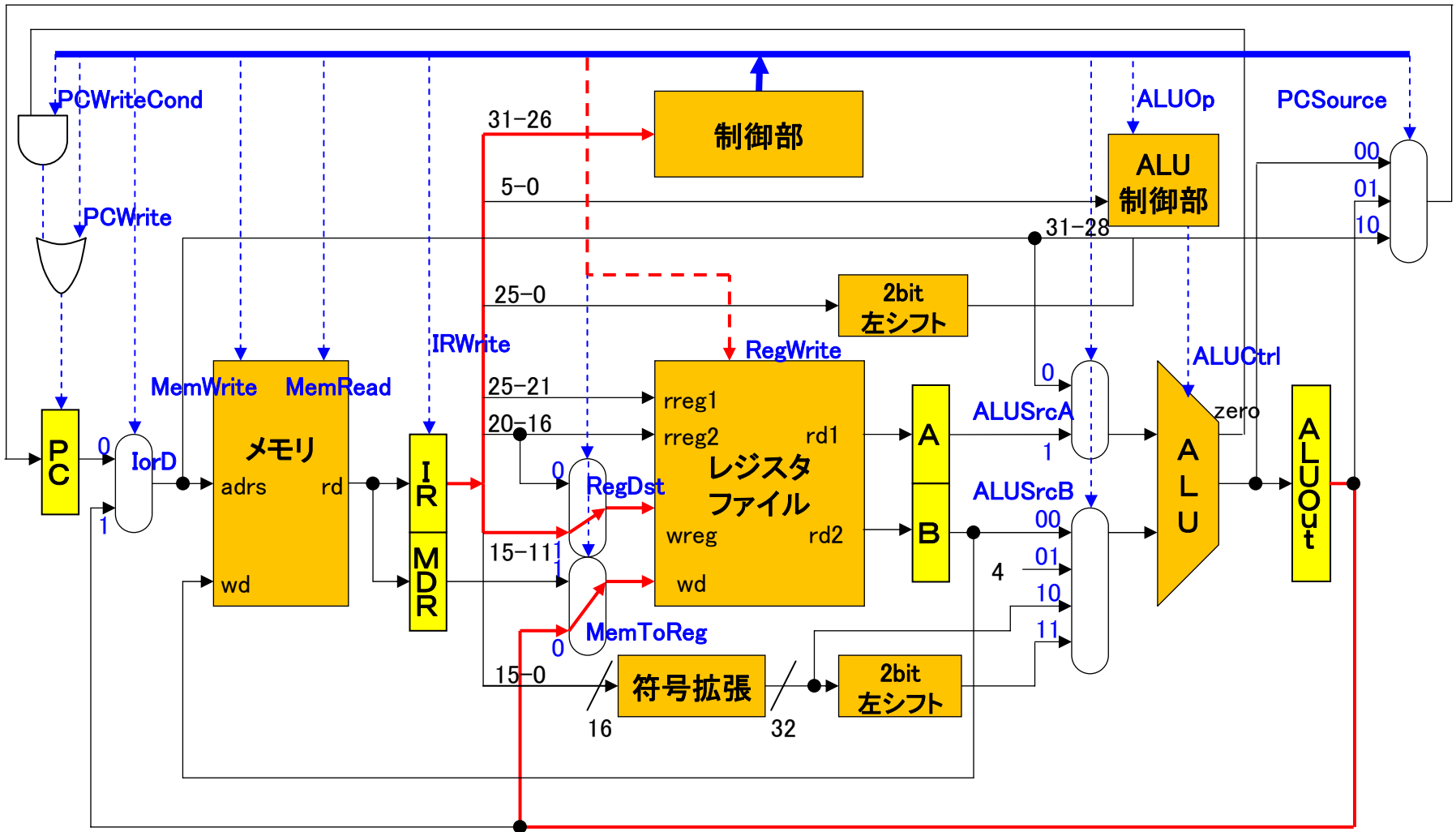
R形式命令の実行(命令解読とレジスタ読出し)



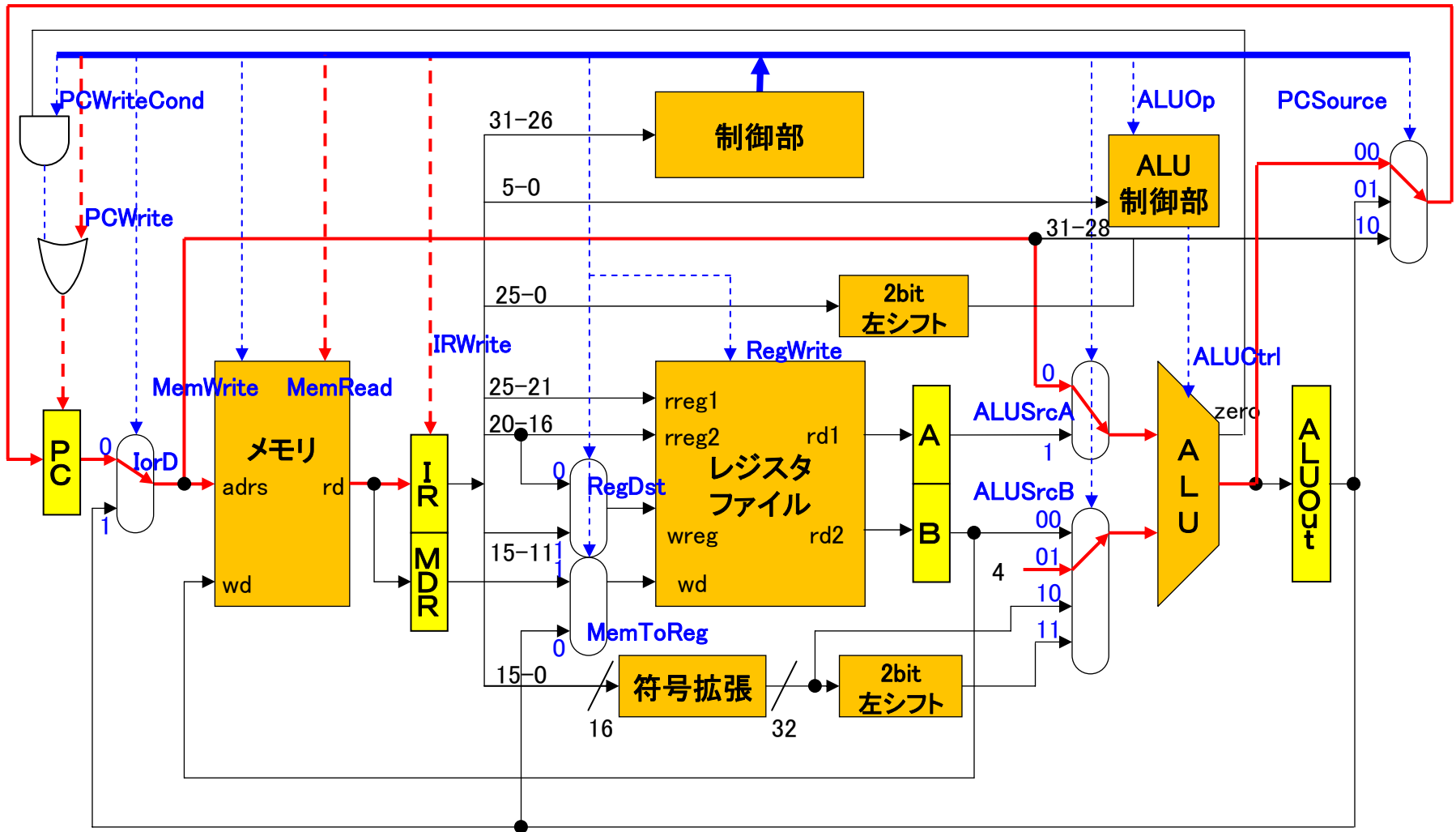
R形式命令の実行(演算の実行)



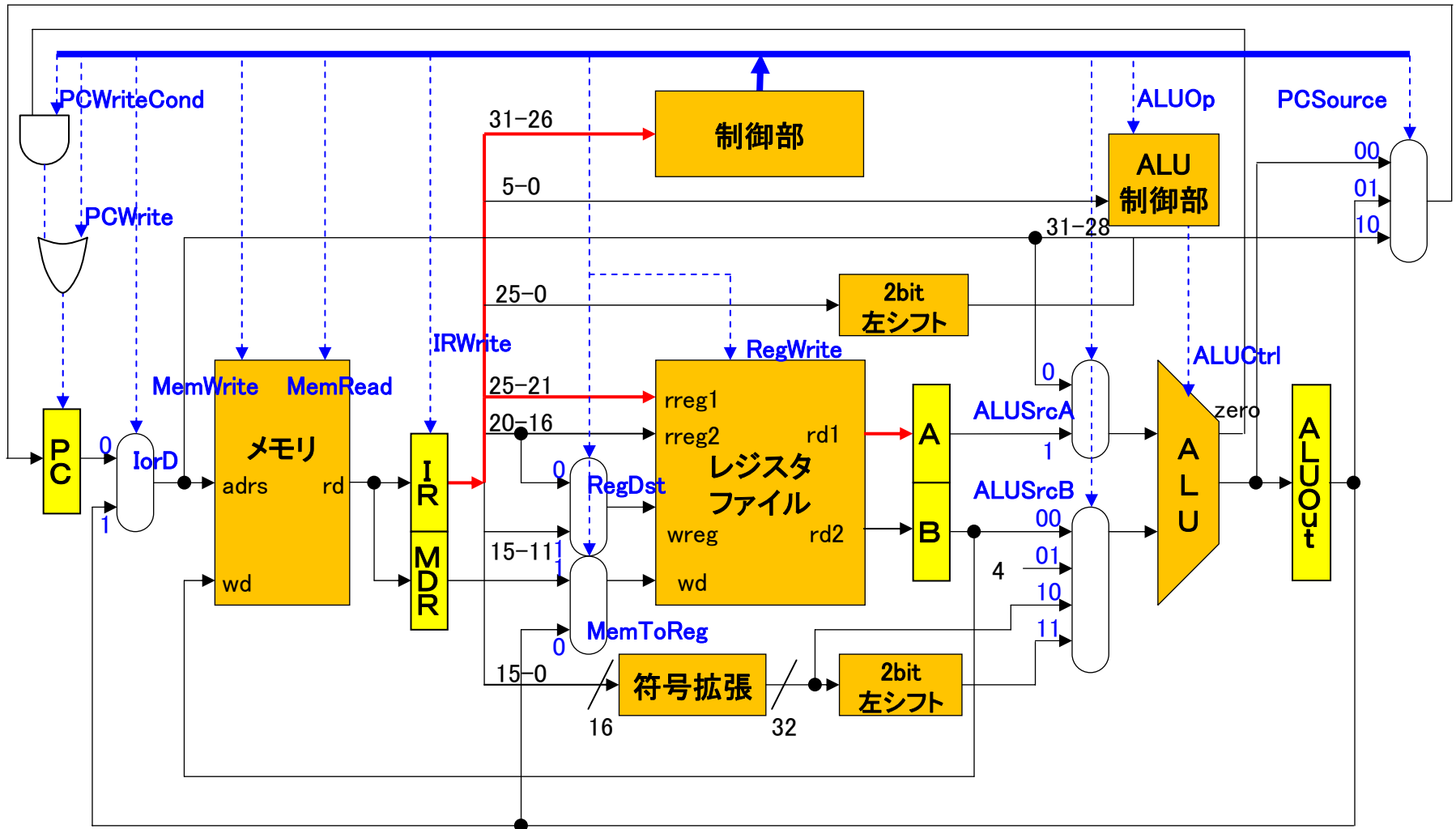
R形式命令の実行(演算結果の書込み)



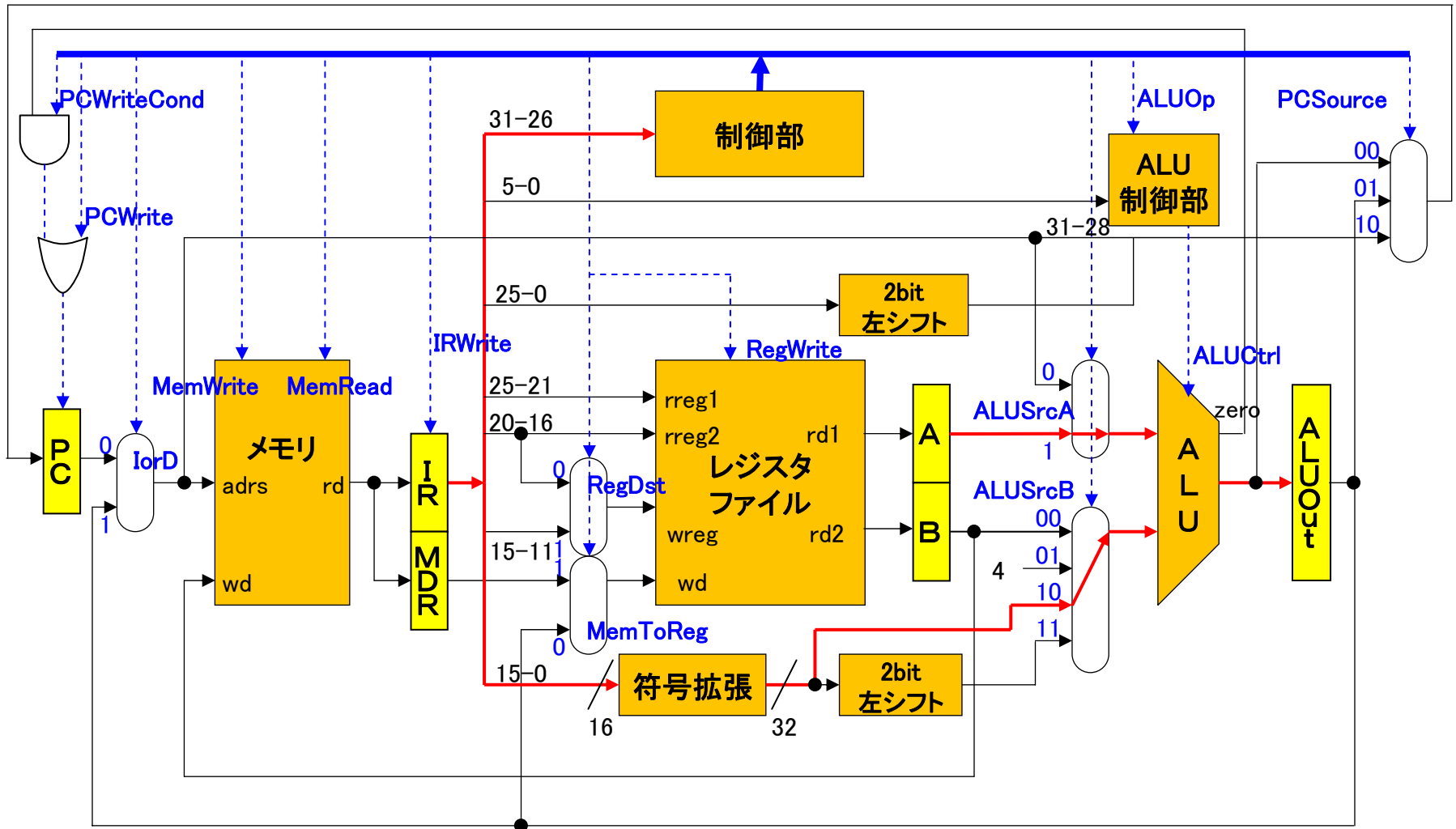
lw命令の実行(命令取得とPC更新)



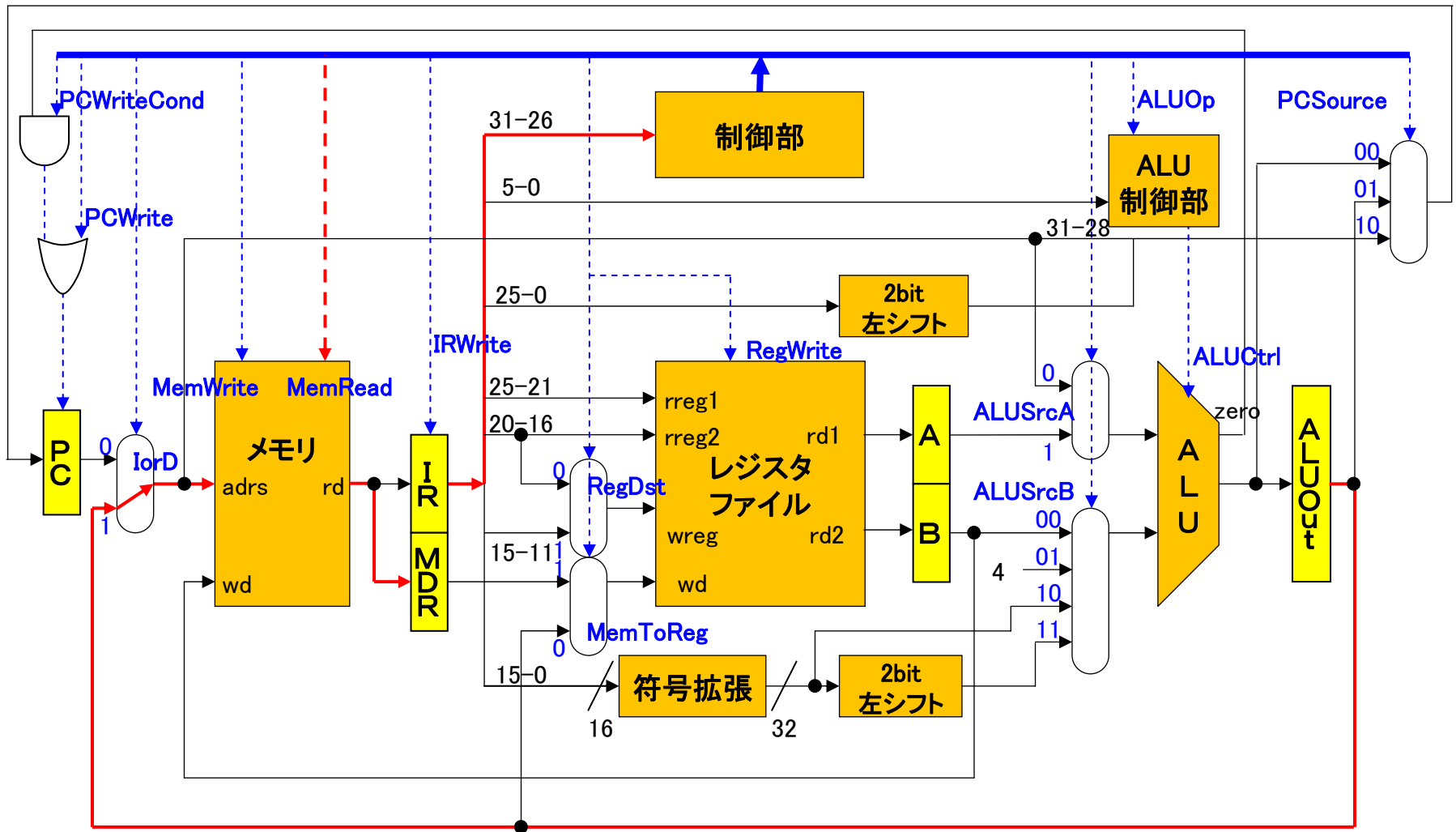
lw命令の実行(命令解読とレジスタ読出し)



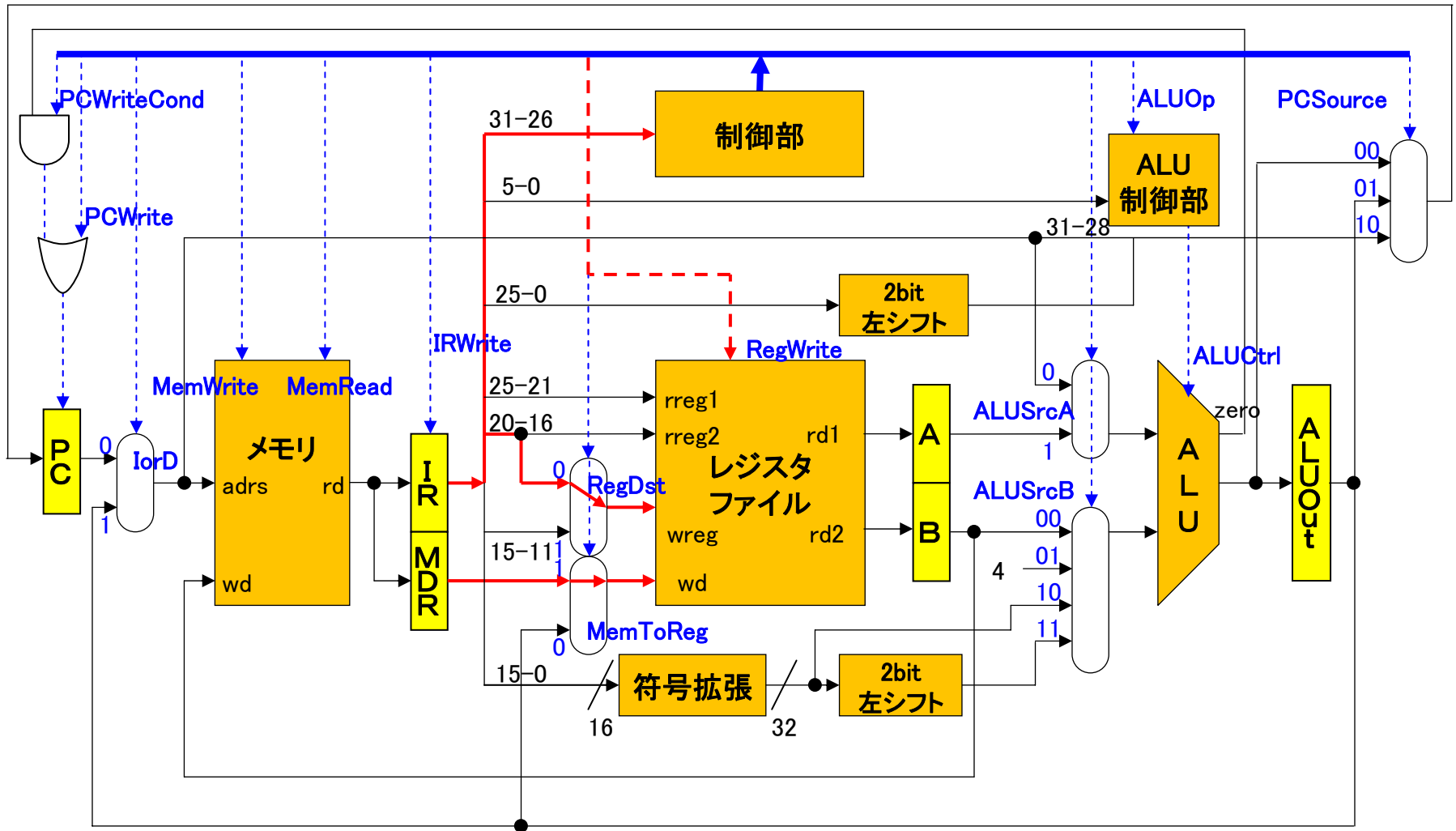
lw命令の実行(アドレス計算)



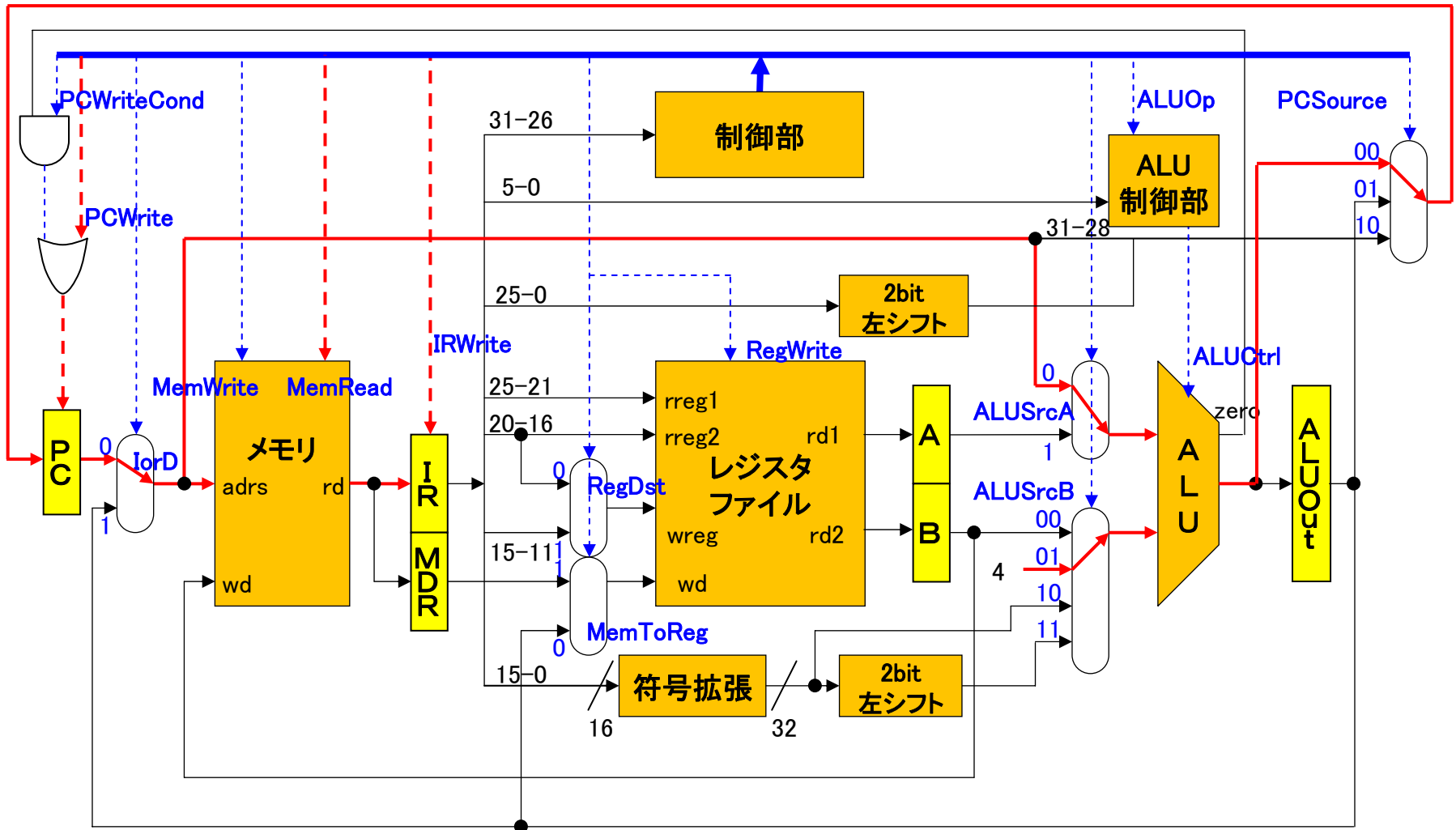
lw命令の実行(メモリへのアクセス)



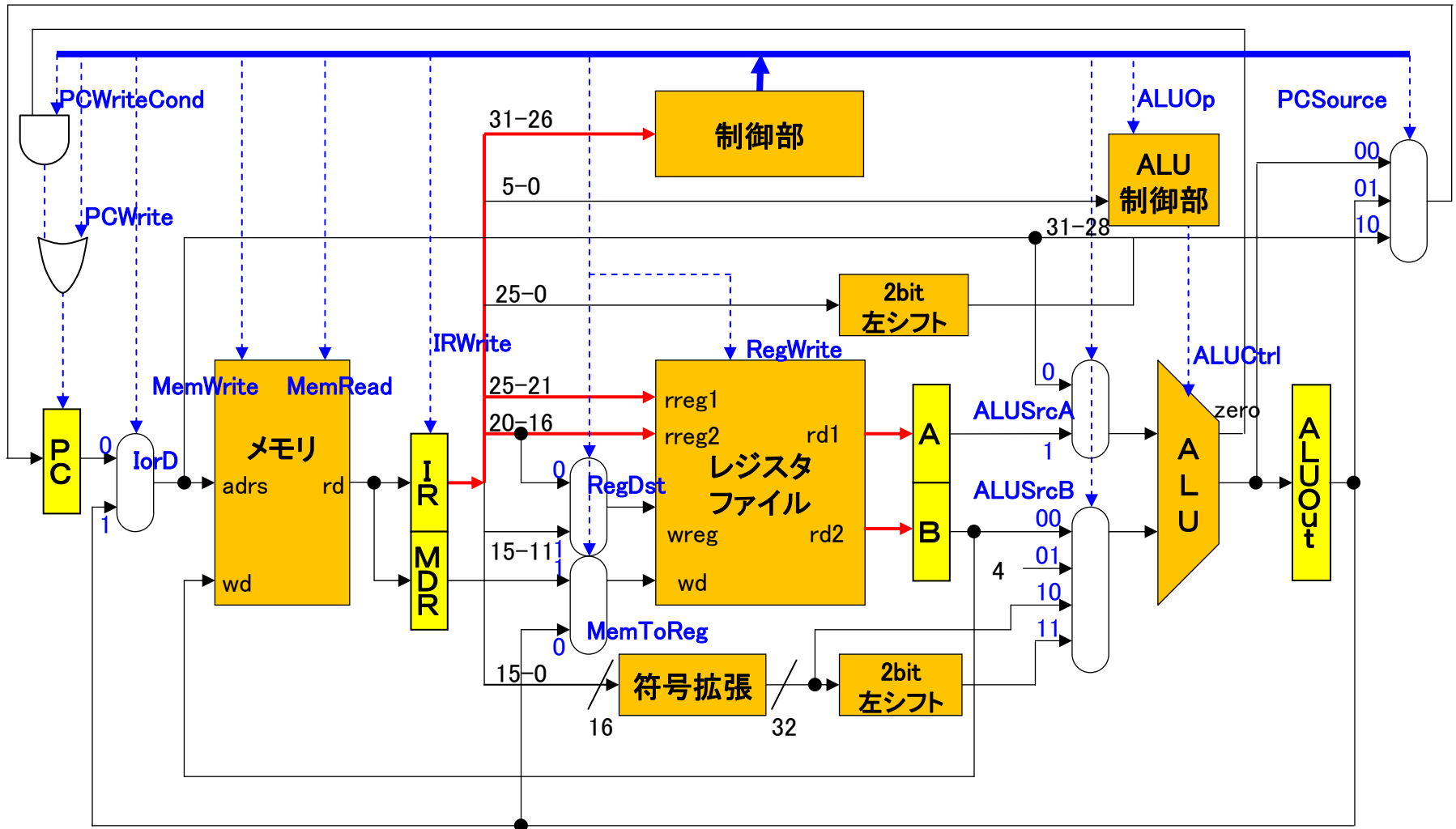
lw命令の実行(ロード結果の書込み)



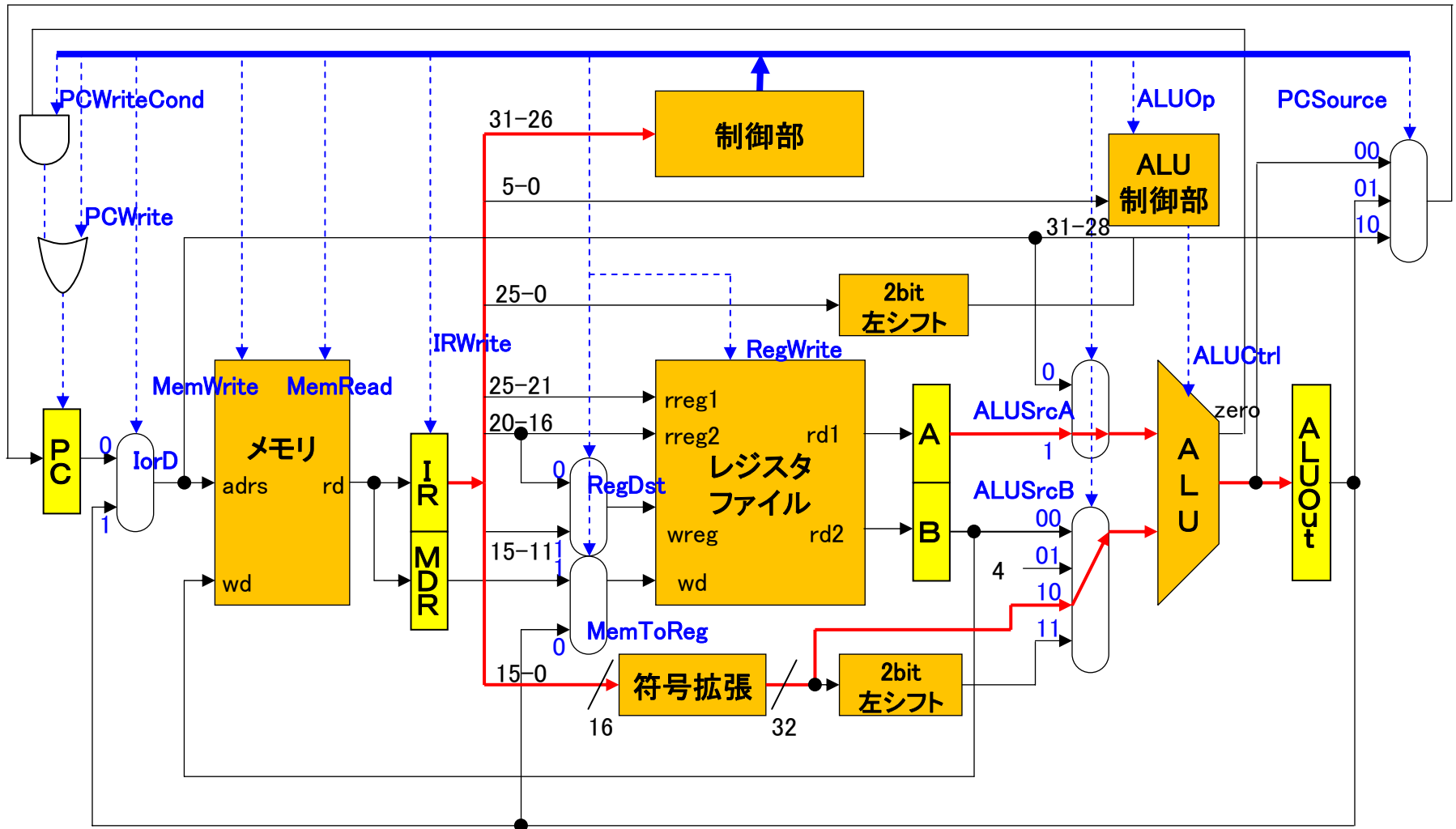
sw命令の実行(命令取得とPC更新)



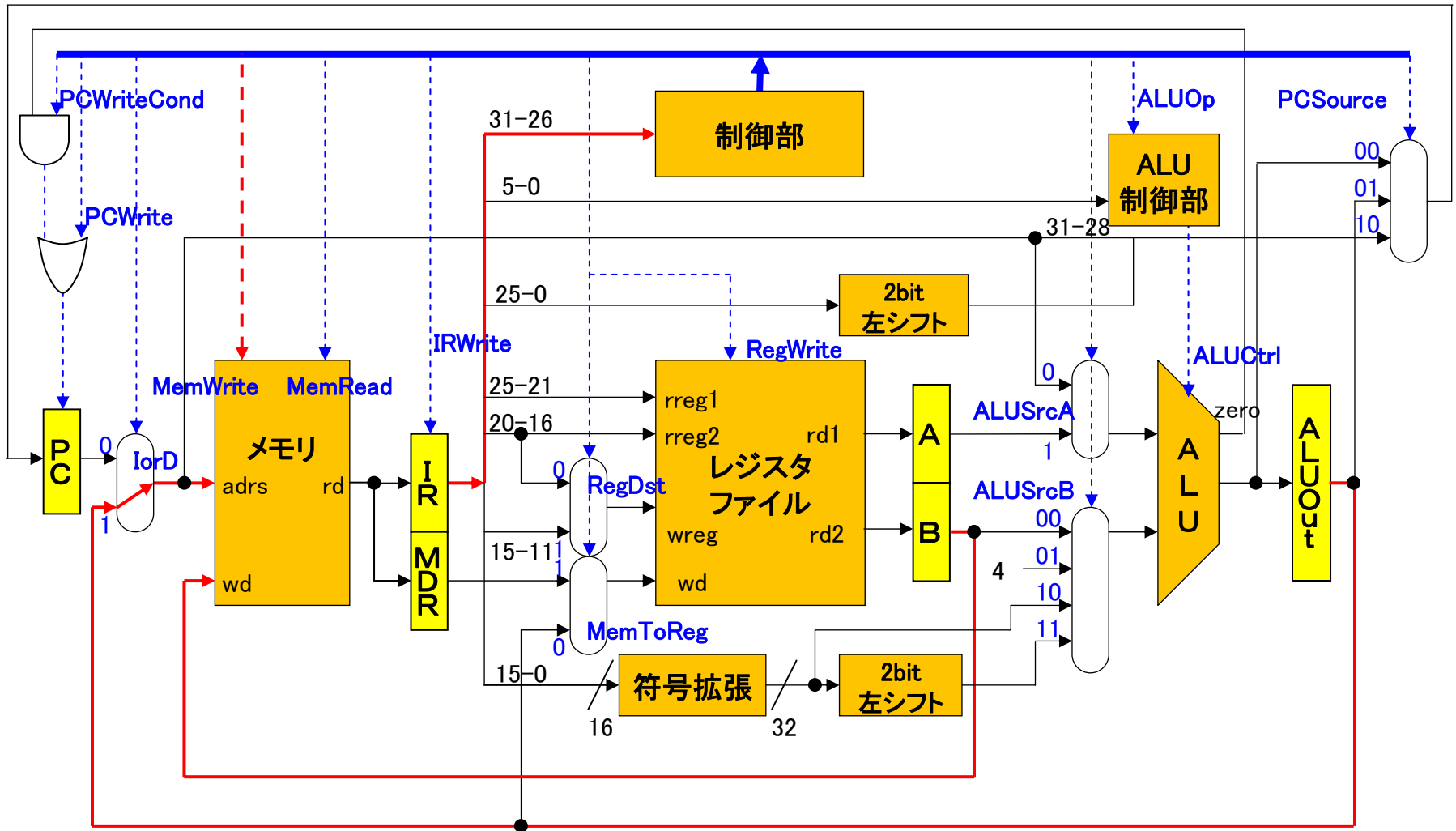
SW命令の実行(命令解釈とレジスタ読出し)



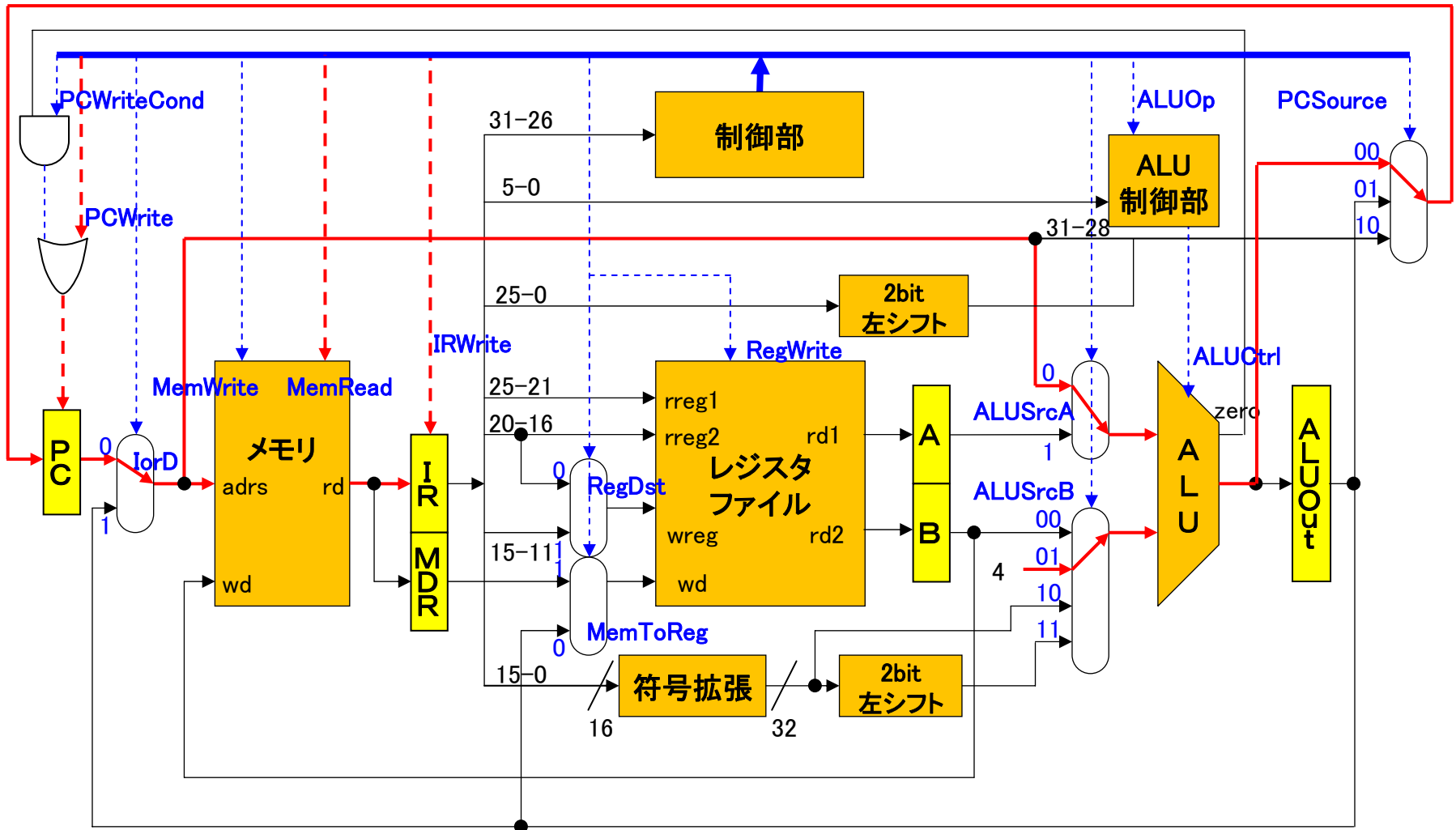
SW命令の実行(アドレス計算)



SW命令の実行(メモリへのアクセス)

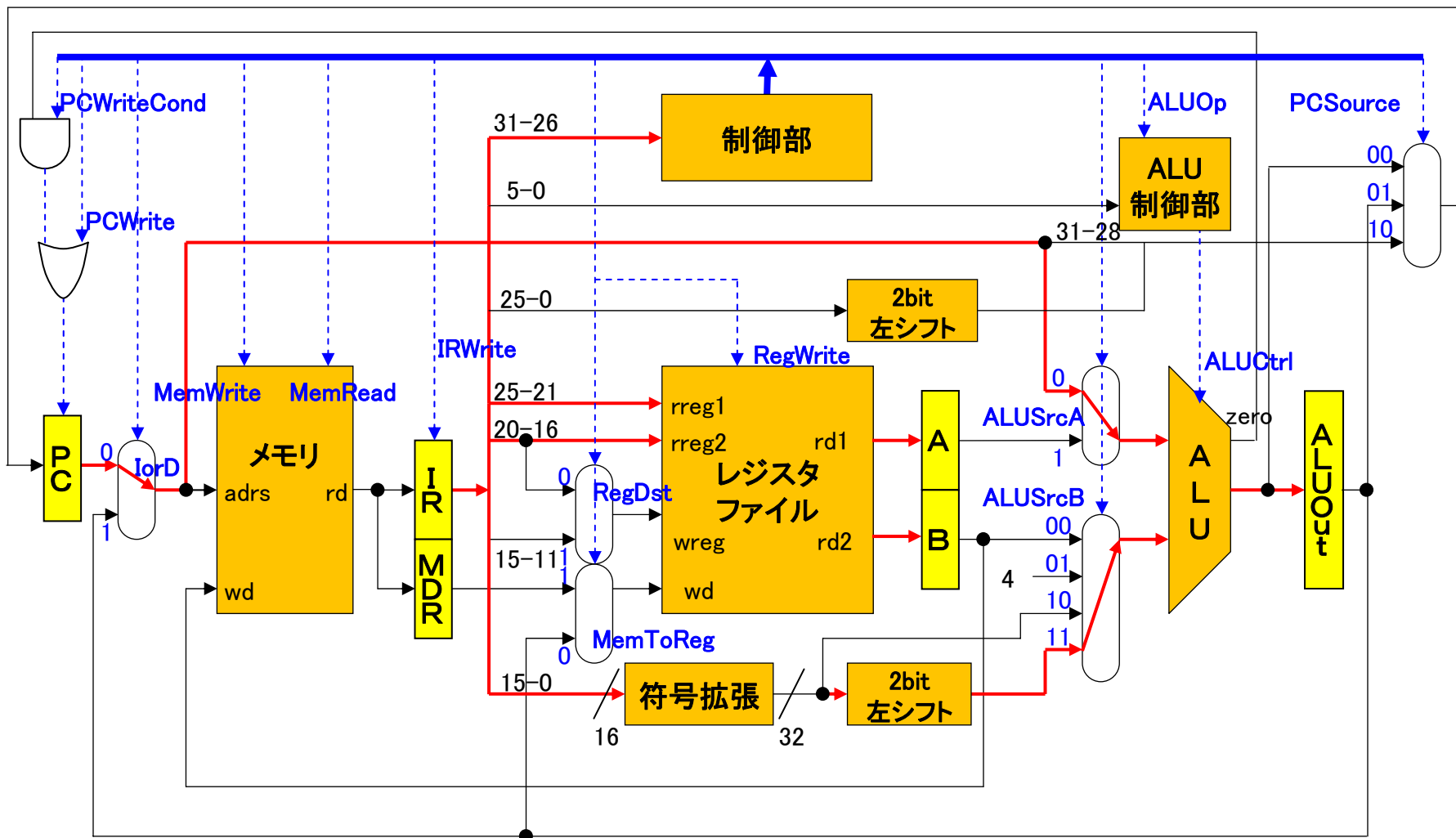


beq命令の実行(命令取得とPC更新)

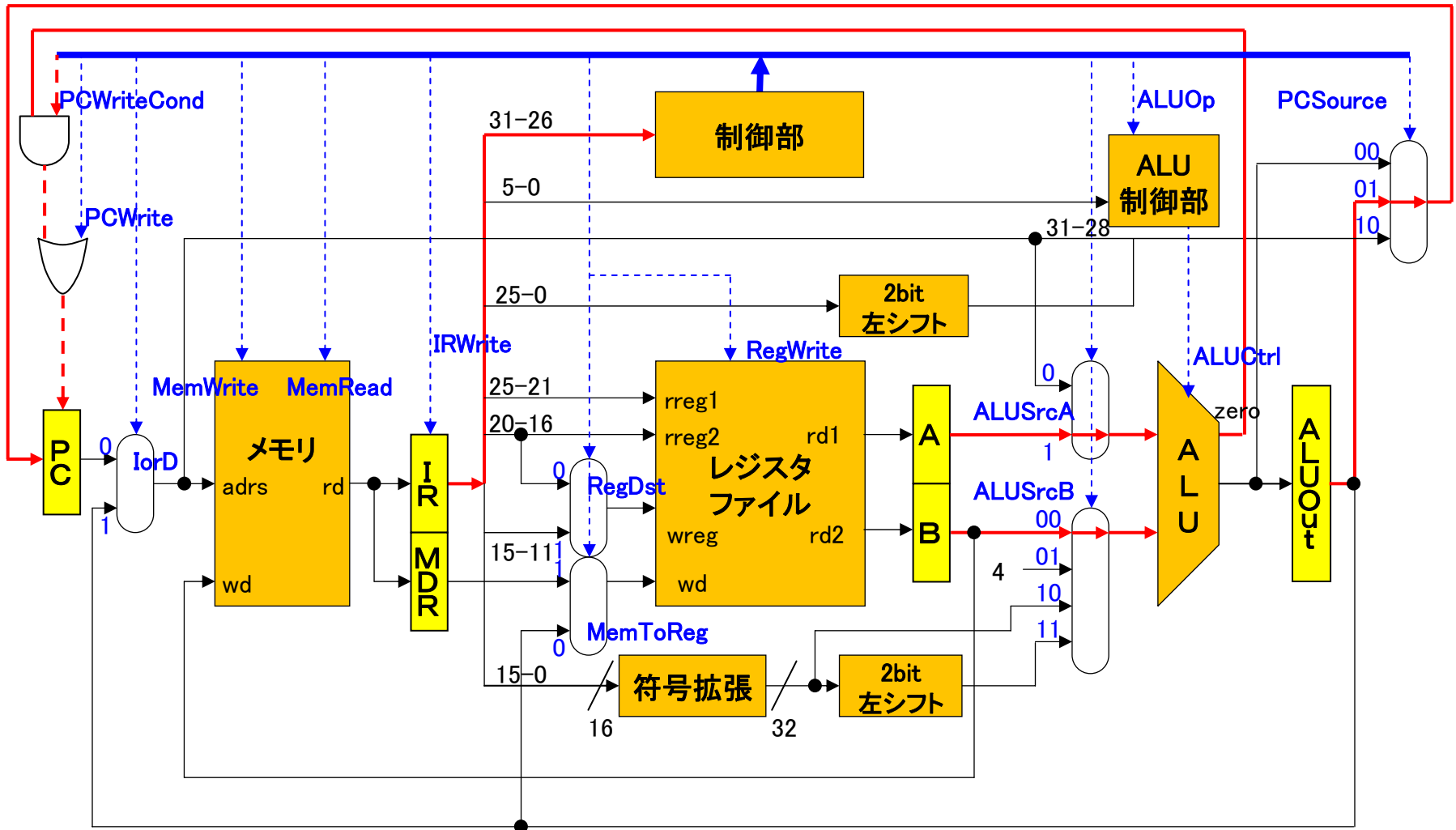


beq命令の実行

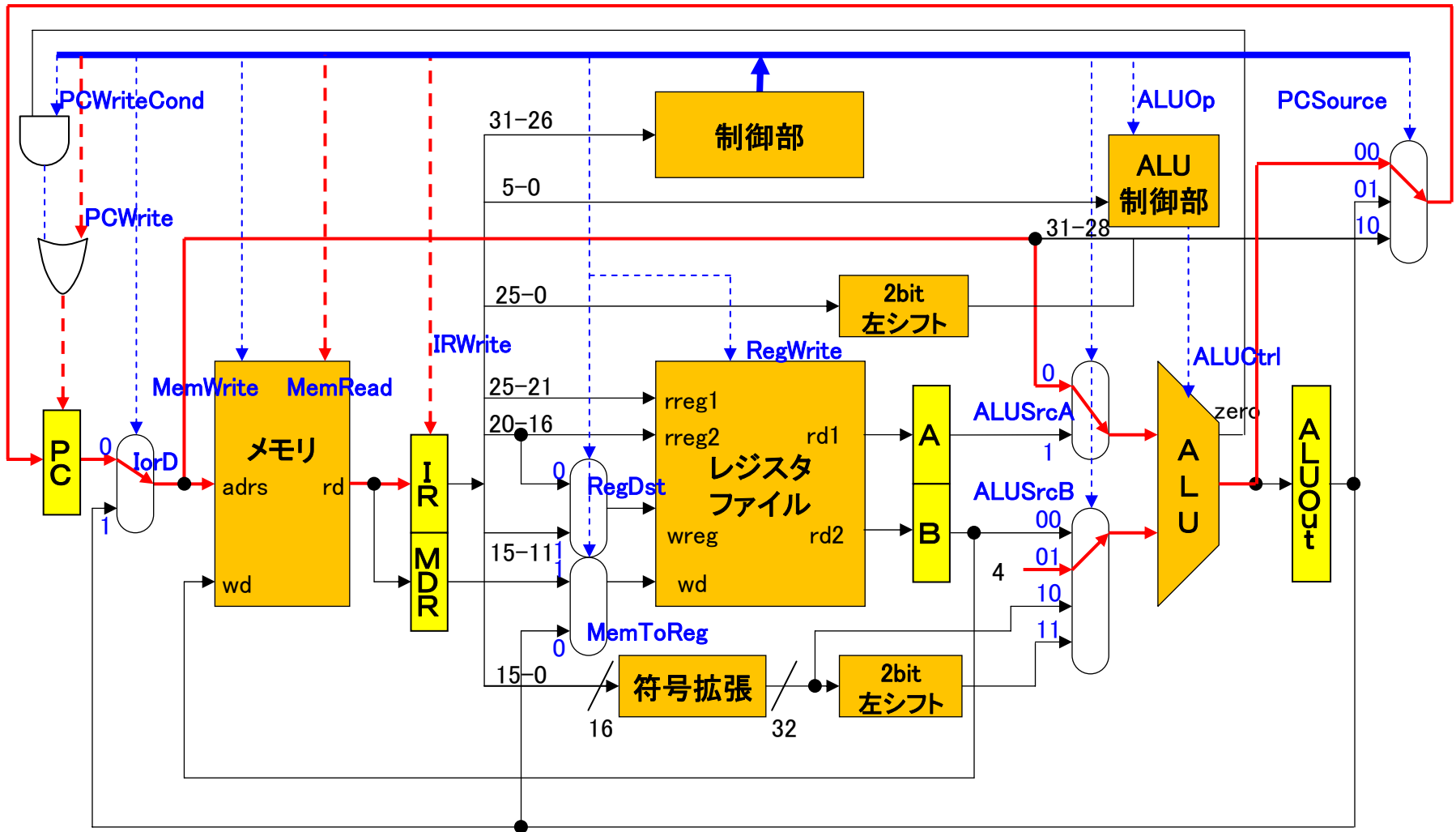
(命令解釈とレジスタ読出し, 分岐先PCの計算)



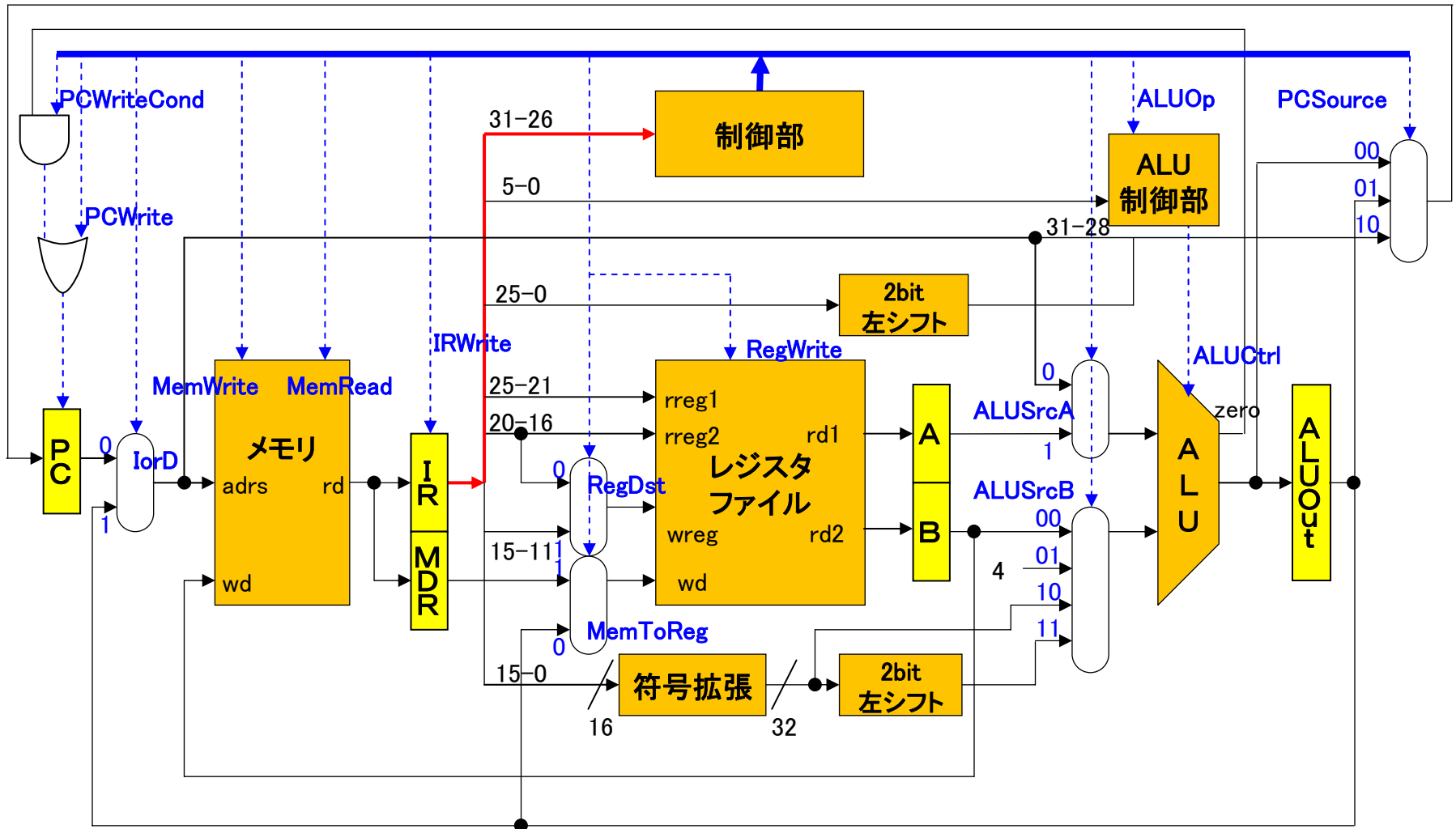
beq命令の実行(分岐条件判定とPC更新)



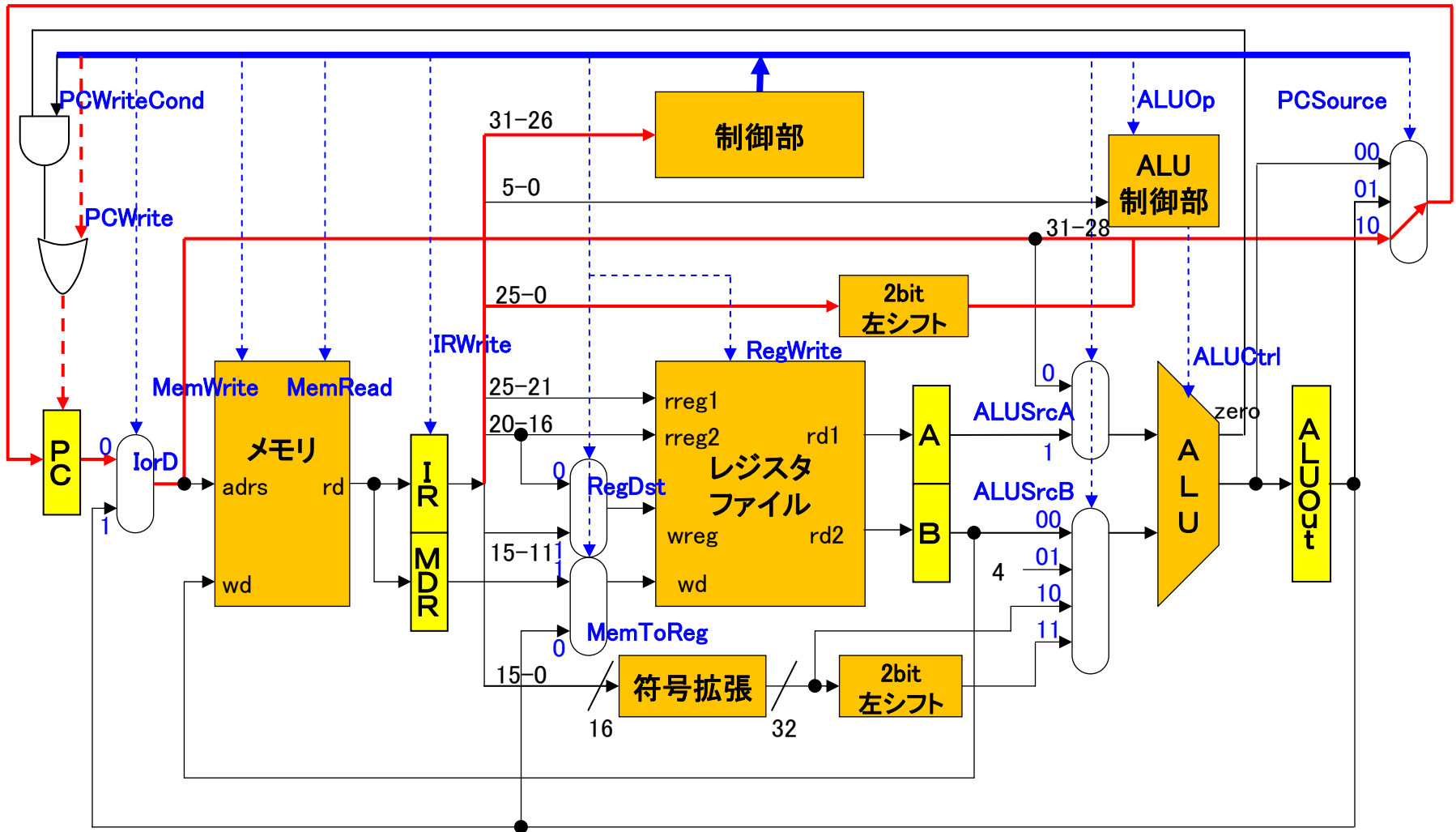
j命令の実行(命令取得とPC更新)



命令の実行(命令の解読)



j命令の実行(PC更新)



制御部の設計(1)

	R形式	lw	sw	beq	j
第1サイクル	IorD=0; MemRead=1; IRWrite=1; ALUSrcA=0; ALUSrcB=01; ALUOp=00; PCSource=00; PCWrite=1				
第2サイクル				ALUSrcA=0 ALUSrcB=11 ALUOp=00	
第3サイクル	ALUSrcA=1 ALUSrcB=00 ALUOp=10	ALUSrcA=1 ALUSrcB=10 ALUOp=00	ALUSrcA=1 ALUSrcB=10 ALUOp=00	ALUSrcA=1 ALUSrcB=00 ALUOp=01 PCSource=01 PCWriteCond=1	PCSource=10 PCWrite=1
第4サイクル	RegDst=1 MemToReg=0 RegWrite=1	IorD=1 MemRead=1	IorD=1 MemWrite=1		
第5サイクル		MemToReg=1 RegDst=0 RegWrite=1			

制御部の設計(2)

	R形式	lw	sw	beq	j
第1サイクル	IorD=0; MemRead=1; IRWrite=1; ALUSrcA=0; ALUSrcB=01; ALUOp=00; PCSource=00; PCWrite=1				
第2サイクル	ALUSrcA=0 ALUSrcB=11 ALUOp=00	ALUSrcA=0 ALUSrcB=11 ALUOp=00	ALUSrcA=0 ALUSrcB=11 ALUOp=00	ALUSrcA=0 ALUSrcB=11 ALUOp=00	ALUSrcA=0 ALUSrcB=11 ALUOp=00
第3サイクル	ALUSrcA=1 ALUSrcB=00 ALUOp=10	ALUSrcA=1 ALUSrcB=10 ALUOp=00	ALUSrcA=1 ALUSrcB=10 ALUOp=00	ALUSrcA=1 ALUSrcB=00 ALUOp=01 PCSource=01 PCWriteCond=1	PCSource=10 PCWrite=1
第4サイクル	RegDst=1 MemToReg=0 RegWrite=1	IorD=1 MemRead=1	IorD=1 MemWrite=1		
第5サイクル		MemToR RegDst=0 RegWrite=1			

共通

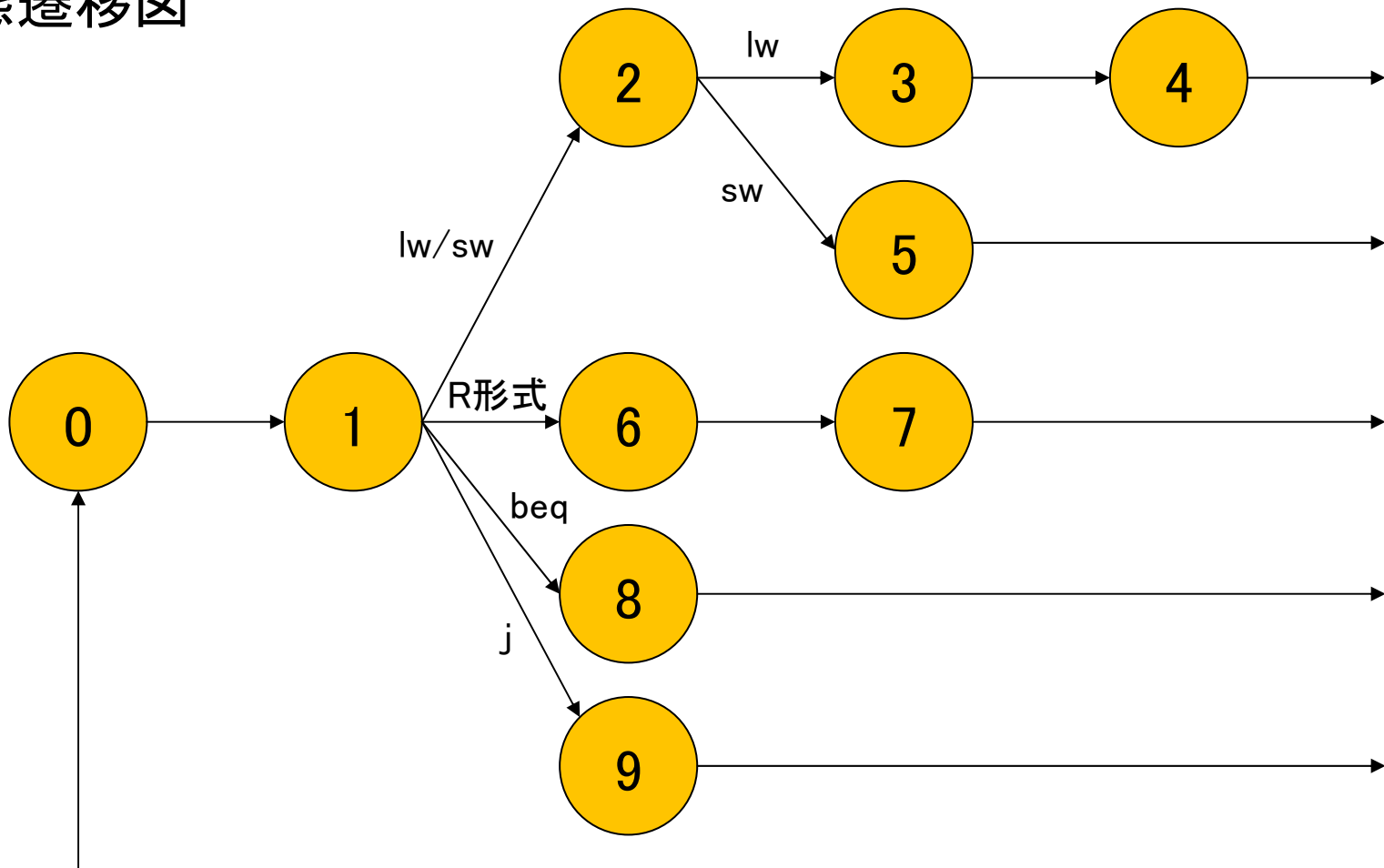
やる必要はないが、
やっても無害。やっ
たほうが制御部が
簡潔になる。

制御部の設計(3)

	R形式	lw	sw	beq	j
第1サイクル	IorD=0; MemRead=1; IRWrite=1; ALUSrcA=0; ALUSrcB=01; ALUOp=00; PCSource=00; PCWrite=1 ①				
第2サイクル	ALUSrcA=0; ALUSrcB=11; ALUOp=00 ①				
第3サイクル	ALUSrcA=1 ALUSrcB=00 ALUOp=10 ⑥	ALUSrcA=1 ALUSrcB=10 ALUOp=00 ②	ALUSrcA=1 ALUSrcB=00 ALUOp=01 ⑧ PCSource=01 PCWriteCond=1 ⑨	PCSource=10 PCWrite=1 ⑨	
第4サイクル	RegDst=1 MemToReg=0 RegWrite=1 ⑦	IorD=1 MemRead=1 ③	IorD=1 MemWrite=1 ⑤		
第5サイクル		MemToReg=1 RegDst=0 RegWrite=1 ④			

制御部の設計(4)

状態遷移図



制御部の設計(5)

制御部

