Microoperaciones a realizar en la implementación propuesta del CS2010

Autores: David Guerrero. Isabel Gómez

1. INSTRUCCIONES DE ACCESO A MEMORIA

Código 0: ST (Rj), Ri

Guarda el registro Ri en la dirección de memoria apuntada por Rj.

CICLO	MEM(Rj) ⟵ Ri	
1	AC ⟵ REG(IR_{2-0})	WAC, OP3, OP2
2	MAR ⟵ AC, AC ⟵ REG(IR_{10-8})	WMAR, RAC, WAC, OP2, OP1
3	MDR ⟵ AC	WMDR, RAC
4	MEM(MAR)⟵MDR	WMEM

Código 1: LD Ri, (Rj)

Guarda en Ri el dato de la dirección de memoria apuntada por Rj.

CICLO	Ri ⟵ MEM(Rj)	
1	AC ⟵ REG(IR_{2-0})	WMAR, OP3, OP2
2	MAR ⟵ AC, **AC ⟵ REG(IR_{10-8})	WMAR, RAC, WAC, OP2, OP1
3	MDR⟵MEM(MAR)	RMEM, WMDR, I/$\overline{O}$MDR
4	REG (IR_{10-8}) ⟵ MDR	WREG, I/OMDR

Código 2: STS dirección, Ri

Guarda el registro Ri en la dirección de memoria apuntada por dirección.

CICLO	MEM(dirección) ⟵ Ri	
1	AC ⟵ IR_{7-0}	WAC, OP3, OP2, INM
2	MAR ⟵ AC, AC ⟵ REG(IR_{10-8})	WMAR, RAC, WAC, OP2, OP1
3	MDR ⟵ AC	WMDR, RAC
4	MEM(MAR)⟵MDR	WMEM

Código 3: LDS Ri, dirección

Guarda en Ri el dato de la dirección de memoria apuntada por dirección.

CICLO	Ri ⟵ MEM(dirección)	
1	AC ⟵ IR_{7-0}	WAC, OP3, OP2, INM
2	MAR ⟵ AC, **AC ⟵ REG(IR_{10-8})	WMAR, RAC, WAC, OP2, OP1
2	MDR⟵MEM(MAR)	RMEM, WMDR, I/OMDR
3	REG (IR_{10-8}) ⟵ MDR	WREG, I/$\overline{O}$MDR

2. INSTRUCCIONES DE SALTO

Código 4: CALL DIRECCION

Salta a la dirección especificada apilando previamente el PC (salto a subrutina).

CICLO	MEM(SP) ⟵ PC, SP ⟵ SP-1, PC ⟵ DIRECCION	
1	MDR ⟵ PC, AC ⟵ IR_{7-0}	WMDR, RPC, WAC, OP3, OP2, INM
2	MAR ⟵ SP, SP ⟵ SP-1	WMAR, RSP, DSP
3	PC ⟵ AC, MEM(MAR)⟵MDR	WPC, RAC, WMEM

Código 5: RET

Desapila un dato escribiéndolo en PC (retornoOP3, OP2. INM de subrutina).

CICLO	PC⟵ MEM (SP+1), SP ⟵ SP+1	
1	SP ⟵ SP+1, **AC ⟵ IR_{7-0}	ISP, WAC, OP3, OP2, INM
2	MAR ⟵ SP	WMAR, RSP
3	MDR⟵MEM(MAR)	RMEM, WMDR, I/$\overline{O}$MDR
4	PC ⟵ MDR	WPC, I/$\overline{O}$MDR

Código 6: BRcc DIRECCION

Realiza una escritura en PC (salto) de forma condicional. “cc” denota una condición lógica de entre las mostradas en la tabla 1.

CICLO	cc: PC ⟵ DIRECCION	
1	AC ⟵ IR_{7-0}	INM,OP3, OP2,WAC
2	cc: PC ⟵ AC	cc: WPCM, cc: RAC

CONDICIÓN	nmónico(s)	notas
Z=1	ZS, EQ	será cierta justo tras realizar la resta A-B si y solo si A=B
C=1	CS, LO	será cierta justo tras realizar la resta A-B si y solo si A<B asumiendo notación base 2 sin signo
V=1	VS	será cierta si y solo si el dato recién calculado no es representable en notación base 2 sin signo
N xor V=1	LT	será cierta justo tras realizar la resta A-B si y solo si A<B asumiendo notación complemento a 2

Tabla 1: Condiciones de salto

Código 7: JMP DIRECCION

Realiza una escritura en PC (salto) de forma incondicional.

CICLO	PC ⟵ DIRECCION	
1	AC ⟵ IR_{7-0}	INM,OP3,OP2,WAC
2	PC ⟵ AC	WPC, RAC

3. INSTRUCCIONES ARITMÉTICO-LÓGICAS, DE DESPLAZAMIENTO E INSTRUCCIÓN DE PARADA

Código 8: ADD Ri, Rj

Suma a un registro el contenido de otro.

CICLO	Ri ⟵ Ri + Rj MOD 2^8, V ⟵ NOT $Ri_{(C2}$ + $Rj_{(C2}$ ∈ [-2^7, 2^7-1], C ⟵ $Ri_{(2}$ + $Rj_{(2}$ > 2^8-1, N ⟵ Ri + Rj MOD 2^8 > 2^7-1, Z ⟵ Ri + Rj = 0 MOD 2^8	
1	AC ⟵ REG(IR_{10-8}) +¸REG(IR_{2-0}) MOD 2^8, V ⟵ NOT REG(IR_{10-8})$_{(C2}$ +¸REG(IR_{2-0})$_{(C2}$ ∈ [-2^7, 2^7-1], C ⟵ REG(IR_{10-8})$_{((2}$ + REG(IR_{2-0})$_{(2}$ > 2^8-1, N ⟵ REG(IR_{10-8}) + REG(IR_{2-0}) MOD 2^8 > 2^7-1, Z ⟵ REG(IR_{10-8}) + REG(IR_{2-0}) = 0 MOD 2^8	WAC, OP3, WS
2	REG(IR_{10-8}) ⟵ AC	WREG, RAC

Código 10: SUB Ri, Rj

Resta a un registro el contenido de otro.

CICLO	Ri ⟵ Ri - Rj MOD 2^8, V ⟵ NOT $Ri_{(C2}$ - $Rj_{(C2}$ ∈ [-2^7, 2^7-1], C ⟵ $Ri_{(2}$ - $Rj_{(2}$ < 0, N ⟵ Ri - Rj MOD 2^8 > 2^7-1, Z ⟵ Ri - Rj = 0 MOD 2^8	
1	AC ⟵ REG(IR_{10-8}) - REG(IR_{2-0}) MOD 2^8, V ⟵ NOT REG(IR_{10-8})$_{(C2}$ -¸REG(IR_{2-0})$_{(C2}$ ∈ [-2^7, 2^7-1], C ⟵ REG(IR_{10-8})$_{((2}$ - REG(IR_{2-0})$_{(2}$ < 0, N ⟵ REG(IR_{10-8}) - REG(IR_{2-0}) MOD 2^8 > 2^7-1, Z ⟵ REG(IR_{10-8}) - REG(IR_{2-0}) = 0 MOD 2^8	WAC, OP3, OP1, WS
2	REG(IR_{10-8}) ⟵ AC	WREG, RAC

Código 11: CP Ri, Rj

Compara el contenido de dos registros.

CICLO	$V \longleftarrow$ NOT $Ri_{(C2}$ - $Rj_{(C2} \in [-2^7, 2^7-1]$, $C \longleftarrow Ri_{(2}$ - $Rj_{(2} < 0$, $N \longleftarrow Ri - Rj \text{ MOD } 2^8 > 2^7-1$, $Z \longleftarrow Ri - Rj = 0 \text{ MOD } 2^8$	
1	**AC $\longleftarrow$ $REG(IR_{10-8}) - REG(IR_{2-0}) \text{ MOD } 2^8$, $V \longleftarrow$ NOT $REG(IR_{10-8})_{(C2}$ -,$REG(IR_{2-0})_{(C2} \in [-2^7, 2^7-1]$, $C \longleftarrow REG(IR_{10-8})_{((2} - REG(IR_{2-0})_{(2} < 0$, $N \longleftarrow REG(IR_{10-8}) - REG(IR_{2-0}) \text{ MOD } 2^8 > 2^7-1$, $Z \longleftarrow REG(IR_{10-8}) - REG(IR_{2-0}) = 0 \text{ MOD } 2^8$	WAC, OP3, OP1, WS

Código 15: MOV Ri, Rj

Almacena en un registro el contenido de otro.

CICLO	$Ri \longleftarrow Rj$	
1	AC$\longleftarrow$ $REG(IR_{2-0})$	WAC, OP3, OP2
2	$REG(IR_{10-8}) \longleftarrow$ AC	RAC, WREG

Código 18: CLC

Pone a 0 el bit de acarreo.

CICLO	$C \longleftarrow 0$	
1	**AC$\longleftarrow$ N.I., $C \longleftarrow 0$	INM, WAC, WS

Código 19: SEC

Pone a 1 el bit de acarreo.

CICLO	$C \longleftarrow 1$	
1	**AC$\longleftarrow$ N.I., $C \longleftarrow 1$	INM, WAC, WS, OP1,OP0

Código 20: ROR Ri

Realiza una rotación hacia la derecha sobre un registro.

CICLO	$Ri \longleftarrow SHR(Ri, C)$, $V \longleftarrow C$ EXOR $Ri_{0,}$ $C \longleftarrow Ri_0$, $N \longleftarrow C$ $Z \longleftarrow$ (NOT C) AND (NOT $OR_{i=1}^{7} R_i$)	
1	AC $\longleftarrow$ $SHR(REG(IR_{10-8}), C)$, $V \longleftarrow C$ EXOR $REG(IR_{10-8})_0$, $C \longleftarrow REG(IR_{10-8})_0$, $N \longleftarrow C$, $Z \longleftarrow$ (NOT C) AND (NOT $OR_{i=1}^{7} REG(IR_{10-8})_i$)	WAC, OP2, INM, WS
2	$REG(IR_{10-8}) \longleftarrow$AC	WREG. RAC

Código 21: ROL Ri

Realiza una rotación hacia la izquierda sobre un registro.

CICLO	$Ri \longleftarrow SHL(Ri, C)$, $V \longleftarrow Ri_7$ EXOR $Ri_{6,}$ $C \longleftarrow Ri_7$, $N \longleftarrow Ri_6$, $Z \longleftarrow$ (NOT C) AND (NOT $OR_{i=0}^{6} R_i$)	
1	$AC \longleftarrow SHL(REG(IR_{10-8}), C)$, $V \longleftarrow REG(IR_{10-8})_7$ EXOR $REG(IR_{10-8})_6$, $C \longleftarrow REG(IR_{10-8})_7$, $N \longleftarrow REG(IR_{10-8})_6$, $Z \longleftarrow$ (NOT C) AND (NOT $OR_{i=0}^{6} REG(IR_{10-8})_i$)	WAC, OP2, OP0, INM, WS
2	$REG(IR_{10-8}) \longleftarrow AC$	WREG. RAC

Código 23: STOP

Detiene la ejecución del programa y lleva el procesador al estado de espera.

Código 24: ADDI Ri, K

Suma a un registro una constante K que se da como operando.

CICLO	$Ri \longleftarrow Ri + K \text{ MOD } 2^8$, $V \longleftarrow$ NOT $Ri_{(C2} + K_{(C2} \in [-2^7, 2^7-1]$, $C \longleftarrow Ri_{(2} + K_{(2} > 2^8-1$, $N \longleftarrow Ri + K \text{ MOD } 2^8 > 2^7-1$, $Z \longleftarrow Ri + K = 0 \text{ MOD } 2^8$	
1	$AC \longleftarrow REG(IR_{10-8}) + IR_{7-0}$ $V \longleftarrow$ NOT $REG(IR_{10-8})_{(C2} +_{,} IR_{7-0(C2} \in [-2^7, 2^7-1]$, $C \longleftarrow REG(IR_{10-8})_{((2} + IR_{7-0(2} > 2^8-1$, $N \longleftarrow REG(IR_{10-8}) + IR_{7-0} \text{ MOD } 2^8 > 2^7-1$, $Z \longleftarrow REG(IR_{10-8}) + IR_{7-0} = 0 \text{ MOD } 2^8$	WAC, OP3, INM, WS
2	$REG(IR_{10-8}) \longleftarrow AC$	WREG, RAC

Código 26: SUBI Ri, K

Resta a un registro una constante K que se da como operando.

CICLO	$Ri \longleftarrow Ri - K \text{ MOD } 2^8$, $V \longleftarrow$ NOT $Ri_{(C2} - K_{(C2} \in [-2^7, 2^7-1]$, $C \longleftarrow Ri_{(2} - K_{(2} < 0$, $N \longleftarrow Ri - K \text{ MOD } 2^8 > 2^7-1$, $Z \longleftarrow Ri - K = 0 \text{ MOD } 2^8$	
1	$AC \longleftarrow REG(IR_{10-8}) - IR_{7-0}$ $V \longleftarrow$ NOT $REG(IR_{10-8})_{(C2} -_{,} IR_{7-0(C2} \in [-2^7, 2^7-1]$, $C \longleftarrow REG(IR_{10-8})_{((2} - IR_{7-0(2} < 0$, $N \longleftarrow REG(IR_{10-8}) - IR_{7-0} \text{ MOD } 2^8 > 2^7-1$, $Z \longleftarrow REG(IR_{10-8}) - IR_{7-0} = 0 \text{ MOD } 2^8$	WAC, OP3, OP1, INM, WS
2	$REG(IR_{10-8}) \longleftarrow AC$	WREG, RAC

Código 27: CPI Ri, K

Compara el contenido de registro con un dato inmediato.

CICLO	$V \longleftarrow NOT\ Ri_{(C2} - K_{(C2} \in [-2^7, 2^7-1]$, $C \longleftarrow Ri_{(2} - K_{(2} > 2^8-1$, $N \longleftarrow Ri - K\ MOD\ 2^8 > 2^7-1$, $Z \longleftarrow Ri - K = 0\ MOD\ 2^8$	
1	**$AC \longleftarrow REG(IR_{10-8}) - IR_{7-0}$ $V \longleftarrow NOT\ REG(IR_{10-8})_{(C2} - IR_{7-0(C2} \in [-2^7, 2^7-1]$, $C \longleftarrow REG(IR_{10-8})_{((2} - IR_{7-0(2} < 0$, $N \longleftarrow REG(IR_{10-8}) - IR_{7-0}\ MOD\ 2^8 > 2^7-1$, $Z \longleftarrow REG(IR_{10-8}) - IR_{7-0} = 0\ MOD\ 2^8$	WAC, OP3, OP1, INM, WS

Código 31: LDI Ri, K

Almacena en un registro una constante K que se da como operando.

CICLO	$Ri \longleftarrow K$	
1	$AC \longleftarrow IR_{7-0}$	WAC, OP3, OP2, INM
2	$REG(IR_{10-8}) \longleftarrow AC$	WREG, RAC

**: Estas transferencias entre registros no son necesarias para la realización de las instrucciones correspondientes. Se incluyen para la optimizar el diseño de la unidad de control.