

Microoperaciones a realizar en la implementación propuesta del CS2010

Autores: David Guerrero. Isabel Gómez

Usted es libre de copiar, distribuir y comunicar públicamente la obra y de hacer obras derivadas siempre que se cite la fuente y se respeten las condiciones de la licencia Attribution-Share alike de Creative Commons.

Texto completo de la licencia: <http://creativecommons.org/licenses/by-nc-sa/3.0/es/>

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	WMAR, RAC
3	MDR ← MEM(MAR)	RMEM, WMDR, I/O'MDR
4	REG (IR ₁₀₋₈) ← MDR	WREG, I/O'MDR

Código 2: STS dirección, ir

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	WMAR, RAC
2	MDR ← MEM(MAR)	RMEM, WMDR, I/O'MDR
3	REG (IR ₁₀₋₈) ← MDR	WREG, I/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 (retorno OP3, OP2. INM de subrutina).

CICLO	PC ← MEM (SP+1), SP ← SP+1	
1	SP ← SP+1,	ISP
2	MAR ← SP	WMAR, RSP
3	MDR ← MEM(MAR)	RMEM, WMDR, I/O'MDR
4	PC ← MDR	WPC, I/O'MDR

Código 6: BRcc DIRECCION

Realiza una escritura en PC (salto) de forma condicional. "cc" denota una condición lógica de entre las 8 mostradas en la tabla de condiciones de salto

CICLO	cc: PC ← DIRECCION	
1	AC ← IR ₇₋₀	INM, OP3, OP2, WAC
2	cc: PC ← AC	cc: WPCM, cc: RAC

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

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) ∈ [-2⁷, 2⁷-1], C ← Ri₍₂₎ + Rj_{(2) > 2⁸-1, N ← Ri + Rj MOD 2⁸ > 2⁷-1, Z ← Ri + Rj = 0 MOD 2⁸}}	
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) ∈ [-2⁷, 2⁷-1], C ← REG(IR₁₀₋₈)₍₂₎ + REG(IR₂₋₀)_{(2) > 2⁸-1, N ← REG(IR₁₀₋₈) + REG(IR₂₋₀) MOD 2⁸ > 2⁷-1, Z ← REG(IR₁₀₋₈) + REG(IR₂₋₀) = 0 MOD 2⁸}}	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) ∈ [-2⁷, 2⁷-1], C ← Ri₍₂₎ - Rj_{(2) > 2⁸-1, N ← Ri - Rj MOD 2⁸ > 2⁷-1, Z ← Ri - Rj = 0 MOD 2⁸}}	
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) ∈ [-2⁷, 2⁷-1], C ← REG(IR₁₀₋₈)₍₂₎ - REG(IR₂₋₀)_{(2) > 2⁸-1, N ← REG(IR₁₀₋₈) - REG(IR₂₋₀) MOD 2⁸ > 2⁷-1, Z ← REG(IR₁₀₋₈) - REG(IR₂₋₀) = 0 MOD 2⁸}}	WAC, OP3, OP1 WS
2	REG(IR ₁₀₋₈) ← AC	WREG, RAC

Código 11: CP Ri, Rj

Compara el contenido de dos registros.

CICLO	$R_i - R_j \text{ MOD } 2^8, V \leftarrow \text{NOT } R_{i(C2)} - R_{j(C2)} \in [-2^7, 2^7-1],$ $C \leftarrow R_{i(2)} - R_{j(2)} > 2^8-1, N \leftarrow R_i - R_j \text{ MOD } 2^8 > 2^7-1, Z \leftarrow R_i - R_j = 0 \text{ MOD } 2^8$	
1	$AC \leftarrow \text{REG}(\text{IR}_{10-8}) - \text{REG}(\text{IR}_{2-0}) \text{ MOD } 2^8,$ $V \leftarrow \text{NOT } \text{REG}(\text{IR}_{10-8})_{(C2)} - \text{REG}(\text{IR}_{2-0})_{(C2)} \in [-2^7, 2^7-1],$ $C \leftarrow \text{REG}(\text{IR}_{10-8})_{(2)} - \text{REG}(\text{IR}_{2-0})_{(2)} > 2^8-1,$ $N \leftarrow \text{REG}(\text{IR}_{10-8}) - \text{REG}(\text{IR}_{2-0}) \text{ MOD } 2^8 > 2^7-1,$ $Z \leftarrow \text{REG}(\text{IR}_{10-8}) - \text{REG}(\text{IR}_{2-0}) = 0 \text{ MOD } 2^8$	WAC, OP3, OP1 WS

Código 15: MOV Ri, Rj

Almacena en un registro el contenido de otro.

CICLO	$R_i \leftarrow R_j$	
1	$AC \leftarrow \text{REG}(\text{IR}_{2-0})$	WAC, OP3, OP2
2	$\text{REG}(\text{IR}_{10-8}) \leftarrow AC$	RAC, WREG

Código 18: CLC

Pone a 0 el bit de acarreo.

CICLO	$C \leftarrow 0$	
1	$AC \leftarrow \text{N.I.}$ $C \leftarrow 0$	INM, WAC WS

Código 19: SEC

Pone a 1 el bit de acarreo.

CICLO	$C \leftarrow 1$	
1	$AC \leftarrow \text{N.I.}$ $C \leftarrow 1$	INM, WAC WS, OP1, OP0

Código 20: ROR Ri

Realiza una rotación hacia la derecha sobre un registro.

CICLO	$R_i \leftarrow \text{SHR}(R_i, C), V \leftarrow C \text{ EXOR } R_{i0}, C \leftarrow R_{i0}, N \leftarrow C$ $Z \leftarrow (\text{NOT } C) \text{ AND } (\text{NOT } \text{OR}_{i=1}^7 R_i)$	
1	$AC \leftarrow \text{SHR}(\text{REG}(\text{IR}_{10-8}), C),$ $V \leftarrow C \text{ EXOR } \text{REG}(\text{IR}_{10-8})_0, C \leftarrow \text{REG}(\text{IR}_{10-8})_0,$ $N \leftarrow C, Z \leftarrow (\text{NOT } C) \text{ AND } (\text{NOT } \text{OR}_{i=1}^7 \text{REG}(\text{IR}_{10-8})_i)$	WAC, OP2, INM WS
2	$\text{REG}(\text{IR}_{10-8}) \leftarrow AC$	WREG. RAC

Código 21: ROL Ri

Realiza una rotación hacia la izquierda sobre un registro.

CICLO	$R_i \leftarrow \text{SHL}(R_i, C), V \leftarrow R_{i7} \text{ EXOR } R_{i6}, C \leftarrow R_{i7}, N \leftarrow R_{i6}$ $Z \leftarrow (\text{NOT } C) \text{ AND } (\text{NOT } \text{OR}_{i=0}^6 R_i)$	
1	$AC \leftarrow \text{SHL}(\text{REG}(\text{IR}_{10-8}), C),$ $V \leftarrow \text{REG}(\text{IR}_{10-8})_7 \text{ EXOR } \text{REG}(\text{IR}_{10-8})_6,$ $C \leftarrow \text{REG}(\text{IR}_{10-8})_7, N \leftarrow \text{REG}(\text{IR}_{10-8})_6,$ $Z \leftarrow (\text{NOT } C) \text{ AND } (\text{NOT } \text{OR}_{i=0}^6 \text{REG}(\text{IR}_{10-8})_i)$	WAC, OP2, OP0, INM WS
2	$\text{REG}(\text{IR}_{10-8}) \leftarrow AC$	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	$R_i \leftarrow R_i + K \text{ MOD } 2^8, V \leftarrow \text{NOT } R_{i(C2)} + K_{(C2)} \in [-2^7, 2^7-1],$ $C \leftarrow R_{i(2)} + K_{(2)} > 2^8-1, N \leftarrow R_i + K \text{ MOD } 2^8 > 2^7-1, Z \leftarrow R_i + K = 0 \text{ MOD } 2^8$	
1	$AC \leftarrow \text{REG}(\text{IR}_{10-8}) + \text{IR}_{7-0}$ $V \leftarrow \text{NOT } \text{REG}(\text{IR}_{10-8})_{(C2)} + \text{IR}_{7-0(C2)} \in [-2^7, 2^7-1],$ $C \leftarrow \text{REG}(\text{IR}_{10-8})_{(2)} + \text{IR}_{7-0(2)} > 2^8-1,$ $N \leftarrow \text{REG}(\text{IR}_{10-8}) + \text{IR}_{7-0} \text{ MOD } 2^8 > 2^7-1,$ $Z \leftarrow \text{REG}(\text{IR}_{10-8}) + \text{IR}_{7-0} = 0 \text{ MOD } 2^8$	WAC, OP3, INM WS
2	$\text{REG}(\text{IR}_{10-8}) \leftarrow AC$	WREG, RAC

Código 26: SUBI Ri, K

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

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

Código 27: CPI Ri, K

Compara el contenido de registro con un dato inmediato.

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

Código 31: LDI Ri, K

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

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