

## Apèndix A1.

# TAULA RESUM DEL REPERTORI D'INSTRUCCIONS DEL PROCESSADOR EDUP12

La taula mostra el conjunt d'instruccions emprat en el processador EduP12. Per a cada instrucció la taula mostra la següent informació:

- Joc d'instruccions classificades per tipologia
- N<sup>o</sup> de paraules que ocupa de la instrucció
- Activació del registre d'estat davant l'execució de la instrucció
- Número de cicles d'execució de la instrucció.

La nomenclatura seguida en la taula és la següent:

- Rd, Rs: registres destí i font
- X, Y, Z: registres X, Y i Z. Registres específics d'adreçament corresponents als registres R29, R30 i R31, respectivament.
- C, Z, N, V, I: Bits de carreteig, zero, negatiu, excés i interrupció
- K, k: constant en adreçament i constant genèrica
- b: bit d'estat
- PC: Comptador de programes
- SP: apuntador a pila

Instrucció			Bits d'estat	Codificació	Paraules	Cicles
Aritmètico-lògiques de doble registre						
MOV Rd, Rs	$Rd \leftarrow Rs$	Moure	-	0000 01sd dddd ssss	1	3
AND Rd, Rs	$Rd \leftarrow Rd \wedge Rs$	I-lògica	C, Z, N, V	0010 00sd dddd ssss	1	3
ADC Rd, Rs	$Rd \leftarrow Rd + Rs + C$	Suma amb carreteig	C, Z, N, V	0000 11sd dddd ssss	1	3
ADD Rd, Rs	$Rd \leftarrow Rd + Rs$	Suma	C, Z, N, V	0000 10sd dddd ssss	1	3
CP Rd, Rs	$\leftarrow Rd - Rs$	Comparar	C, Z, N, V	0001 10sd dddd ssss	1	3
CPC Rd, Rs	$\leftarrow Rd - (Rs + C)$	Comparar amb carreteig	C, Z, N, V	0001 11sd dddd ssss	1	3
EOR Rd, Rs	$Rd \leftarrow Rd \oplus Rs$	Or-exclusiva lògica	C, Z, N, V	0010 10sd dddd ssss	1	3
OR Rd, Rs	$Rd \leftarrow Rd \vee Rs$	Or lògica	C, Z, N, V	0010 01sd dddd ssss	1	3
SBC Rd, Rs	$Rd \leftarrow Rd - (Rs + C)$	Resta amb carreteig	C, Z, N, V	0001 01sd dddd ssss	1	3
SUB Rd, Rs	$Rd \leftarrow Rd - Rs$	Resta	C, Z, N, V	0001 00sd dddd ssss	1	3
TST Rd, Rs	$\leftarrow Rd \wedge Rs$	Test bit a bit	C, Z, N, V	0000 01sd dddd ssss	1	3
Aritmètico-lògiques de registre simple						
ASR Rd	$Rd \leftarrow Rd(11) \& Rd(11...1), C \leftarrow Rd(0)$	Desplaçament aritmètic a la dreta	C, Z, N, V	0011 --0d dddd 1100	1	3
CLR Rd	$Rd \leftarrow 0$ (EOR Rd, Rd)	Posar a 0	C, Z, N, V	EOR Rd, Rd	1	3
COM Rd	$Rd \leftarrow (2^n - 1) - Rd$	Complementar (C1)	C, Z, N, V	0011 --0d dddd 1010	1	3
DEC Rd	$Rd \leftarrow Rd - 1$	Decrementar	C, Z, N, V	0011 --0d dddd 1001	1	3
INC Rd	$Rd \leftarrow Rd + 1$	Incrementar	C, Z, N, V	0011 --0d dddd 1000	1	3
LSL Rd	$Rd \leftarrow Rd(10...0) \& 0, C \leftarrow Rd(11) - ADD$	Desplaçament lògic a l'esquerra	C, Z, N, V	ADD Rd, Rd	1	3
LSR Rd	$Rd \leftarrow 0 \& Rd(11...1), C \leftarrow Rd(0)$	Desplaçament lògic a la dreta	C, Z, N, V	0011 --0d dddd 1101	1	3
NEG Rd	$Rd \leftarrow 2^n - Rd$	Negar (C2)	C, Z, N, V	0011 --0d dddd 1011	1	3
ROL Rd	$Rd \leftarrow Rd(10...0) \& C, C \leftarrow Rd(11)$	Rotació a l'esquerra	C, Z, N, V	0011 --0d dddd 1111	1	3
ROR Rd	$Rd \leftarrow C \& Rd(11...1), C \leftarrow Rd(0)$	Rotació a dreta	C, Z, N, V	0011 --0d dddd 1110	1	3
SWAP Rd	$Rd \leftarrow Rd(3...0) \& Rd(11...4)$	Rotació de 4 bits a la dreta sense carry	Z, N, V	0011 -1 d dddd ----	1	3

Operacions amb immediat (doble paraula)						
LDI Rd, k	$Rd \leftarrow k$	Càrrega amb immediat	-	0011 ---d dddd 0000	2	4
ADDI Rd, k	$Rd \leftarrow Rd+k$	Suma amb immediat	C, Z, N, V	0011 ---d dddd 0001	2	4
ANDI Rd, k	$Rd \leftarrow Rd \wedge k$	I lògica amb immediat	C, Z, N, V	0011 ---d dddd 0010	2	4
CPI Rd, k	$- \leftarrow Rd-k$	Comparar amb immediat	C, Z, N, V	0011 ---d dddd 0110	2	4
ORI Rd, k	$Rd \leftarrow Rd \vee k$	O lògica amb immediat	C, Z, N, V	0011 ---d dddd 0011	2	4
SBCI Rd, k	$Rd \leftarrow Rd-(k+C)$	Resta amb carreteig i immediat	C, Z, N, V	0011 ---d dddd 0101	2	4
SUBI Rd, k	$Rd \leftarrow Rd-k$	Resta amb immediat	C, Z, N, V	0011 ---d dddd 0100	2	4
TSTI Rd, k	$- \leftarrow Rd \wedge k$	I lògica amb immediat sense guarda	C, Z, N, V	0011 ---d dddd 0111	2	4
Instruccions de salt condicional						
BRBC b, k	Si $SR(s)=0$ aleshores $PC \leftarrow PC+k+1$	Salt si bit b del SR és 1		1111 kkkk kkkk kbbb	1	3
BRBS b, k	Si $SR(s)=1$ aleshores $PC \leftarrow PC+k+1$	Salt si bit b del SR és 0		1110 kkkk kkkk kbbb	1	3
BRCC k	Si $C=0$ aleshores $PC \leftarrow PC+k+1$	Salt si carreteig és 0		1111 kkkk kkkk k000	1	3
BRCS k	Si $C=1$ aleshores $PC \leftarrow PC+k+1$	Salt si carreteig és 1		1110 kkkk kkkk k000	1	3
...BRSH k	Si $C=0$ aleshores $PC \leftarrow PC+k+1$	Salt si igual o major		1111 kkkk kkkk k000	1	3
...BRLO k	Si $C=1$ aleshores $PC \leftarrow PC+k+1$	Salt si menor		1110 kkkk kkkk k000	1	3
...BRNZ k, BRNE k	Si $Z=0$ aleshores $PC \leftarrow PC+k+1$	Salt si diferent		1111 kkkk kkkk k001	1	3
...BRZE k, BREQ k	Si $Z=1$ aleshores $PC \leftarrow PC+k+1$	Salt si igual		1110 kkkk kkkk k001	1	3
...BRPL k	Si $N=0$ aleshores $PC \leftarrow PC+k+1$	Salt si positiu		1111 kkkk kkkk k010	1	3
...BRMI k	Si $N=1$ aleshores $PC \leftarrow PC+k+1$	Salt si negatiu		1110 kkkk kkkk k010	1	3
BRVC k	Si $V=0$ aleshores $PC \leftarrow PC+k+1$	Salt si excés és 0		1111 kkkk kkkk k011	1	3
BRVS k	Si $V=1$ aleshores $PC \leftarrow PC+k+1$	Salt si excés és 1		1110 kkkk kkkk k011	1	3
BRID k	Si $I=0$ aleshores $PC \leftarrow PC+k+1$	Salt si interrupció inhabilitada		1111 kkkk kkkk k100	1	3
BRIE k	Si $I=1$ aleshores $PC \leftarrow PC+k+1$	Salt si interrupció habilitada		1110 kkkk kkkk k100	1	3

Instruccions de salt incondicional						
ICALL z (o y o x)	Pila $\leftarrow$ PC, PC $\leftarrow$ Z, SP $\leftarrow$ SP +1	Crida indirecte a subrutina		0110 zyx- ---- 0011	1	4
IJMP z (o y o x)	PC $\leftarrow$ Z	Salt indirecte		0110 zyx- ---- 0010	1	3
RCALL k	Pila $\leftarrow$ PC, PC $\leftarrow$ PC + k+1, SP $\leftarrow$ SP +1	Crida relativa a subrutina		0101 kkkk kkkk kkkk	1	4
RJMP k	PC $\leftarrow$ PC + k+1	Salt relatiu		0100 kkkk kkkk kkkk	1	3
RET	SP $\leftarrow$ SP-1, PC $\leftarrow$ pila	Retorn de subrutina		0110 000- ---- 0100	1	4
RETI	SP $\leftarrow$ SP-1, PC $\leftarrow$ pila	Retorn d'interrupció		0110 000- ---- 0101	1	4
Instruccions de càrrega amb memòria de programa						
LPM Rd, Z (o Y o X)	Rd $\leftarrow$ mem(Z), (o Y o X)	Càrrega indirecta		1100 zyx d dddd 1100	1	4
LPM Rd, +Z (o Y o X)	Z $\leftarrow$ Z+1, Rd $\leftarrow$ mem(Z), (o Y o X)	Càrrega indirecta amb pre-increment		1100 zyx d dddd 1101	1	4
LPM Rd, -Z (o Y o X)	Z $\leftarrow$ Z-1, Rd $\leftarrow$ mem(Z), (o Y o X)	Càrrega indirecta amb pre-decrement		1100 zyx d dddd 1111	1	4
Instruccions de càrrega/guarda amb memòria de dades						
LDS Rd, K	Rd $\leftarrow$ (K)	Càrrega directa (doble paraula)		1100 ---d dddd 0100	2	5
LD Rd, Z (o Y o X)	Rd $\leftarrow$ mem(Z), (o Y o X)	Càrrega indirecta		1100 zyx d dddd 0000	1	4
LD Rd, +Z (o Y o X)	Z $\leftarrow$ Z+1, Rd $\leftarrow$ mem(Z), (o Y o X)	Càrrega indirecta amb pre-increment		1100 zyx d dddd 0001	1	4
LD Rd, -Z (o Y o X)	Z $\leftarrow$ Z-1, Rd $\leftarrow$ mem(Z), (o Y o X)	Càrrega indirecta amb pre-decrement		1100 zyx d dddd 0011	1	4
LDD Rd, Z+q	Rd $\leftarrow$ mem(Z+q)	Càrrega indirecte amb desplaçament en Z		1000 qq qd dddd qq qq	1	4
LDD Rd, Y+q	Rd $\leftarrow$ mem(Y+q)	Càrrega indirecte amb desplaçament en Y		1010 qq qd dddd qq qq	1	4
STS K, Rs	mem(K) $\leftarrow$ Rs	Guarda directa (doble paraula)		1101 ---d dddd 0100	2	5
ST Z (o Y o X), Rs	mem(Z) $\leftarrow$ Rs, (o Y o X)	Guarda indirecte		1101 zyx d dddd 0000	1	4
ST +Z (o Y o X), Rs	Z $\leftarrow$ Z+1, mem(Z) $\leftarrow$ Rs, (o Y o X)	Guarda indirecta amb pre-increment		1101 zyx d dddd 0001	1	4
ST -Z (o Y o X), Rs	Z $\leftarrow$ Z-1, mem(Z) $\leftarrow$ Rs, (o Y o X)	Guarda indirecta amb pre-decrement		1101 zyx d dddd 0011	1	4
STD Z+q, Rs	(Z+q) $\leftarrow$ Rs	Guarda indirecte amb desplaçament en Z		1001 qq qd dddd qq qq	1	4
STD Y+q, Rs	(Y+q) $\leftarrow$ Rs	Guarda indirecte amb desplaçament en Y		1011 qq qd dddd qq qq	1	4

Instruccions d'entrada/sortida						
IN Rd, PORT	$Rd \leftarrow \text{PORT}$	Entrada de port		0111 0AA d dddd AAAA	1	3
OUT PORT, Rs	$\text{PORT} \leftarrow Rs$	Sortida a port		0111 1AA d dddd AAAA	1	3
De registre de status						
BSET b	$\text{RegistreEstat}(b) \leftarrow 1$	Posar a 1 el bit <i>b</i> del registre d'estat		0000 0001 0000 0sss	1	3
BCLR b	$\text{RegistreEstat}(b) \leftarrow 0$	Posar a 1 el bit <i>b</i> del registre d'estat		0000 0001 0000 1sss	1	3
SEI	$\text{RegistreEstat}(7) \leftarrow 1$	Posar a 1 el bit 7 del registre d'estat		0000 0001 0000 0111	1	3
CLI	$\text{RegistreEstat}(7) \leftarrow 0$	Posar a 1 el bit 7 del registre d'estat		0000 0001 0000 1111	1	3
Altres instruccions						
PUSH Rs	$\text{Pila} \leftarrow Rs, SP \leftarrow SP + 1$	Posar valor de registre Rs en pila		0110 000d dddd 0111	1	4
POP Rd	$SP \leftarrow SP - 1, Rd \leftarrow \text{Pila}$	Treure de pila a registre Rd		0110 000d dddd 0110	1	4
SAVE	$\text{Pila} \leftarrow SR, SP \leftarrow SP + 1$	Guardar en pila el registre d'estat		0110 100d dddd 0111	1	4
RESTORE	$SP \leftarrow SP - 1, SR \leftarrow \text{Pila}$	Tornar de pila a registre d'estat		0110 100d dddd 0110	1	4
NOP		No fer res		0000 0000 0000 0000	1	3