

Capítol 6

LLENGUATGE MÀQUINA DE EDUP12

En aquest capítol s'introdueix el repertori d'instruccions del processador EduP12 i els modes d'adreçament amb què treballa.

6.1 El repertori d'instruccions.

El *repertori* o *joc d'instruccions* és el conjunt de comandes que implementa una CPU. El joc d'instruccions sol incloure diferents tipus d'instrucció com aritmètico-lògiques, de transferència de dades, d'entrada/sortida, de salt i d'altres de genèriques.

La instrucció forma part de l'arquitectura del processador ja que porta immersos detalls sobre el cablejat que s'ha d'establir durant la descodificació de la mateixa.

Diferents màquines tenen diferents tipus d'instruccions. Cada màquina té el seu format d'instrucció. Però sempre la interpretació de la instrucció es fa en base a uns camps que identifiquen aspectes del funcionament de la CPU. Aquests camps són:

- La codificació de la instrucció. Tota instrucció té un codi únic que la identifica davant la màquina per a ser interpretada
- Els operands o dades de treball. Ha d'especificar sobre quin tipus de dades treballa: registres, constants, ports.
- Adreçament. La màquina ha de saber on es troben els operands i els ha d'anar a cercar. Cada màquina té els seus modes d'adreçament.

En el processador EduP12 la longitud de la instrucció és de 16 bits. Les instruccions es poden agrupar per categories, mantenint certa uniformitat dintre les categories. A més el repertori d'instruccions és variat i incorpora un conjunt de modes d'adreçament que el fan útil per a aplicacions diverses.

6.2. Modes d'adreçament

EduP12 treballa amb el següent conjunt de modes d'adreçament:

- Adreçament implícit.
- Adreçament immediat.
- Adreçament directe.
- Adreçament indirecte.
- Adreçament relatiu.
- Adreçament indexat.

Una mateixa instrucció pot admetre diferents tipus d'aquests modes d'adreçament. També sol passar que una màquina mescli entre sí aquests modes d'adreçament. Per exemple, es veurà que en EduP12 hi ha instruccions que mesclen els modes d'adreçament indirecte i relatiu.

6.2.1 Adreçament implícit

Es troba en les instruccions de retorn de subrutina (RET, RETI) que han de recuperar de la pila l'adreça de la propera instrucció a executar.

6.2.2 Adreçament immediat

En el processador EduP12 totes les instruccions que es tractaran en el grup d'instruccions aritmètico-lògiques amb immediat duen aquest adreçament. La figura 6.1 mostra un esquema del format d'aquestes instruccions. La dada o *immediat* ve especificada per la constant k de la instrucció.

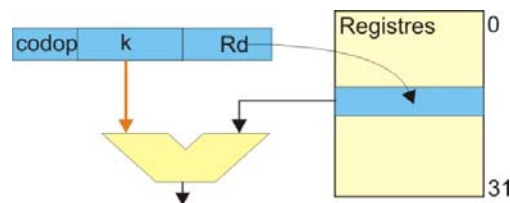


Figura 6.1. Adreçament immediat

6.2.3 Adreçament directe

En EduP12 es poden considerar tres tipus d'adreçament directe:

- Adreçament directe en registre. Es troba en les instruccions que treballen amb dades que es troben en el registre. L'adreça de l'operand és el propi registre. Per exemple, és el cas de la instrucció *ADD R16, R17*: els dos operands estan en registres.
- Adreçament directe en memòria d'entrada/sortida. Quan es treballa amb un port d'entrada/sortida l'adreça de l'operand és el port, que és qui conté la dada amb la que es treballa. Per exemple, en l'execució de la instrucció *IN Rd, PORT*, *PORT* és l'adreça efectiva on hi ha l'operand.
- Adreçament directe en memòria de dades. De fet és el mateix cas de l'adreçament directe amb memòria d'entrada/sortida, només que ara l'adreça efectiva es troba en una segona paraula. Com a exemple en la instrucció *LDS Rd, 0x123*, *0x123* és l'adreça (efectiva) de memòria que conté l'operand (figura 6.2).

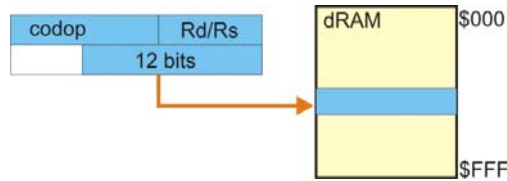


Figura 6.2. Adreçament directe

6.2.4 Adreçament indirecte

EduP12 implementa l'adreçament indirecte per registre. És a dir, un registre conté l'adreça efectiva de l'operand. Els registres d'indirecció només poden ser els registres R29, R30 o R31, anomenats X, Y i Z, respectivament. També admet el pre-increment o pre-decrement del registre.

Exemple: La instrucció LD Rd, -Y realitza una càrrega indirecte amb el registre Y pre-decrementat sobre Rd. Les accions que duu a terme són, així, dues:

Primer decrementa Y: $Y \leftarrow Y-1$.

Després carrega Rd: $Rd \leftarrow \text{dRAM}\langle Y \rangle$

La figura 6.3 mostra un esquema de la indirecció que es duu a terme amb aquesta instrucció.

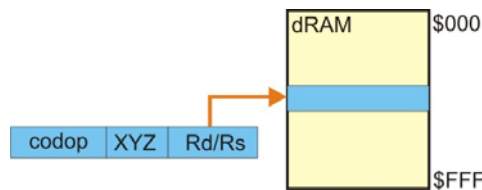


Figura 6.3. Adreçament indirecte

EduP12 també admet un adreçament indirecte a memòria de programa emprant el registre Z que el duu a terme amb les instruccions IJMP i ICALL.

6.2.6 Adreçament relatiu

En EduP12 es poden considerar dos tipus d'adreçament relatiu:

- Adreçament indirecte amb desplaçament. Es suma un desplaçament prefixat a una adreça de referència continguda en un registre (el Y o el Z). És el cas de la instrucció LDD Rd,Y+q, que carrega al registre Rd la dada continguda en la posició Y+q de la memòria de dades (figura 6.4).

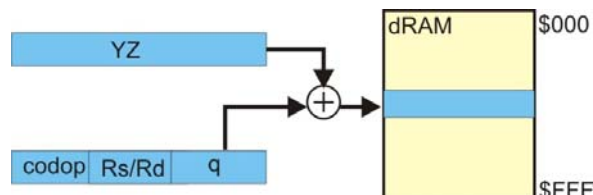


Figura 6.4. Adreçament indirecte relatiu

- Adreçament relatiu a comptador de programes, que es produeix en el cas de les instruccions de salt relatiu. Per exemple, en *RJMP displ*, s'efectua un salt a l'adreça de memòria de programa PC+displ (figura 6.5).

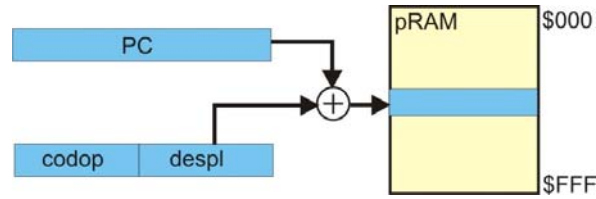


Figura 6.5. Adreçament relatiu a comptador de programes

6.3 Repertori d'instruccions en EduP12

El repertori d'instruccions d' EduP12 es pot classificar en els següents tipus d'instruccions:

- Instruccions aritmètico-lògiques
 - o De doble registre: MOV, ADD, ADC, SUB, SBC, CP, CPC, AND, OR, EOR, TST.
 - o De registre simple: INC, DEC, COM, NEG, CLR, ASR, LSR, LSL, ROR, ROL, SWAP.
 - o Amb immediats: LDI, ADDI, ANDI, ORI, SUBI, SBCI, CPI, TSTI.
- Instruccions de transferència de dades.
 - o Amb memòria de programa.
 - Instruccions de càrrega: LPM, LPM+
 - o Amb memòria de dades.
 - Instruccions de càrrega: LDS, LD, LD+, LD-, LDD
 - Instruccions de guarda: STS, ST, ST+, ST-.
- Instruccions de salt.
 - o De salt incondicional: RJMP, RCALL, RET, RETI.
 - o De salt incondicional indirecte: IJMP, ICALL.
 - o De salt condicional. BRBC, BRBS, BRCC, BRCS, BRSH, BRLO, BRZE, BRNZ, BRMI, BRPL, BRVC, BRVS, BRIE, BRID.
- Instruccions d'entrada/sortida: IN, OUT
- Altres.
 - o De pila o *stack*: POP, PUSH, SAVE, RESTORE.
 - o No fer res: NOP.

A continuació es fa un repàs a les instruccions de eduP12 per tipologia d'instrucció. La codificació particular de cada instrucció es pot trobar en l'apèndix A.

6.3.1 Instruccions aritmètico-lògiques de doble registre

Conjunt d'instruccions

El conjunt d'instruccions aritmètico-lògiques de dos registres inclou el conjunt d'instruccions següent:

- Aritmètiques: ADD, ADC, SUB, SBC,
- Lògiques: AND, OR, EOR
- De comparació de valors: CP, CPC, TST
- De transferència entre registres: MOV

Aquestes instruccions actualitzen el registre d'estat.

Format de les instruccions

El format de les instruccions és del tipus ADD Rd, Rs, que correspon a l'operació $Rd \leftarrow Rd + Rs$.

El codi màquina de la instrucció és:

cccc	ccsd	dddd	ssss
------	------	------	------

En el que

- cccccc correspon a la codificació de les instruccions, que va de 000001 a 001011
- ddddd representa el registre destí, de R0 a R31
- sssss representa el registre font, de R0 a R31

Exemple

ADD R16, R17 queda codificat com 0000 1011 0000 0001 = 0x0B01

6.3.2 Instruccions aritmètico-lògiques de registre simple

Conjunt d'instruccions

Les instruccions aritmètico-lògiques de registre simple treballen amb la dada d'un únic registre. Està format pel conjunt d'instruccions següent:

- Aritmètiques: INC, DEC, NEG.
- Lògiques: COM.
- De desplaçament: ASR, LSR, ROR, ROL, SWAP.

El desplaçament lògic a l'esquerra (LSL) s'implementa via ADD Rd, Rd.

Aquestes instruccions actualitzen el registre d'estat.

Format de les instruccions

El format de les instruccions és del tipus ADDI Rd, k, que correspon a l'operació $Rd \leftarrow Rd + k$.

El codi màquina de la instrucció es compon de dues paraules, i té el format¹:

cccc	--wd	dddd	1ccc
------	------	------	------

I a on:

- cccc-ccc correspon a la codificació de les instruccions, que va de 0011-000 a 0011-111
- ddddd representa el registre destí, de R0 a R31
- kkkk kkkk kkkk kkkk representa la constant amb la que es treballa
- Totes les instruccions tenen el bit w = 0, excepte SWAP que val 1.

Exemple

ADDI R16, 0xF0F queda codificat com

0011 0001 0000 0001 | 0000 1111 0000 1111 = 0x3101 | 0x0F0F

¹ Sempre que aparegui – en un bit indica que aquest valor no està emprat, tot i que en la descodificació s'agafa valor 0.

6.3.3 Instruccions aritmètico-lògiques amb immediats

Conjunt d'instruccions

El conjunt d'instruccions aritmètico-lògiques amb immediats conforma el conjunt d'instruccions que treballen amb constants, i inclou el conjunt d'instruccions següent:

- Aritmètiques: ADDI, SUBI, SBCI
- Lògiques: ANDI, ORI
- De comparació de valors: CPI, TSTI
- De càrrega: LDI

Aquestes instruccions, a excepció de LDI, actualitzen el registre d'estat.

Format de les instruccions

El format de les instruccions és del tipus ADDI Rd, k, que correspon a l'operació $Rd \leftarrow Rd + k$.

El codi màquina de la instrucció es compon de dues paraules, i té el format:

cccc	---d	dddd	0ccc	----	kkkk	kkkk	kkkk
------	------	------	------	------	------	------	------

I a on:

- cccc-ccc correspon a la codificació de les instruccions, que va de 0011-000 a 0011-111
- ddddd representa el registre destí, de R0 a R31
- kkkk kkkk kkkk kkkk representa la constant amb la que es treballa

Exemple

ADDI R16, 0xF0F queda codificat com

0011 0001 0000 0001 | 0000 1111 0000 1111 = 0x3101 | 0xF0F

6.3.4 Instruccions de transferència de dades amb memòria de programa.

La memòria de programes conté el programa que executa el processador.

A nivell de guarda de dades només es permet l'escriptura en el moment de càrrega del programa, pel que pot ser útil per emmagatzemar dades rellevants pel programa, especialment important si són taules de dades.

Per tant, amb la memòria de programa només es permet la lectura de dades.

Conjunt d'instruccions

La instrucció de transferència de dades (de memòria de programa a registre) és:

- Càrrega de programa a registre, només X, Y o Z: LPM Rd, XYZ
- Càrrega de programa a registre, només X, Y o Z amb pre-increment o pre-decrement: LPM Rd, +/-XYZ

Format de les instruccions

- El codi màquina de la instrucció és:

1100	ZYXd	dddd	11ab
------	------	------	------

a on:

- ddddd és el registre
- ZYX val 100, 010 o 001 segons s'empri Z, Y o X, respectivament
- ab és la codificació de la indirecció:
 - 0 → Indirecció normal (sense pre-decrement ni pre-increment)
 - 01 → preincrement
 - 11 → predecrement

Exemple

- LPM R17, -Z. Realitza l'operació
 - ...primer decrementa Z: $Z \leftarrow Z-1$
 - ...i després $\text{mem}\langle Z \rangle \leftarrow R20$

es codifica amb

1100 1001 0001 1111

6.3.5 Instruccions de transferència de dades amb memòria de dades.

Conjunt d'instruccions

El conjunt d'instruccions de transferència de dades transporten les dades de memòria a registre i al revés.

Les instruccions que transfereixen dades de registre a memòria s'anomenen instruccions de guarda (*store*). Les que transfereixen dades de memòria a registre s'anomenen instruccions de càrrega (*load*).

Les instruccions de transferència de dades admeten diferents modes d'adreçament.

El conjunt d'instruccions de transferència de dades és el següent:

- Instruccions de transferència de dades directa.
 - o S'adreça directament la posició de memòria
 - o Necessita d'una segona paraula que conté l'adreça de memòria
 - o Està format per les instruccions LDS i STS.
- Instruccions de transferència de dades indirecta.
 - o La indirecció es realitza via un dels registres X, Y o Z, que corresponen als registres R29, R30 i R31, respectivament.
 - o Els registres X, Y o Z contenen l'adreça (indirecció) a la que s'ha d'anar a cercar la dada.
 - o Hi ha un triple format:
 - Indirecció normal
 - Indirecció amb pre-increment: primer incrementen el registre i després fan la cerca
 - Indirecció amb pre-decrement: primer decrementen el registre i després fan la cerca
 - o El conjunt d'instruccions està format per
 - Indirecció normal: LD XYZ, ST XYZ, on XYZ fa referència a un dels tres registres
 - Indirecció amb pre-increment: LD +XYZ, ST +XYZ
 - Indirecció amb pre-decrement: LD -XYZ, ST -XYZ
- Instruccions de transferència de dades indirecta amb desplaçament
 - o En el moment de fer la indirecció sumen una constant al registre.
 - o Els dos registres permesos per fer la indirecció són Y i Z.
 - o El conjunt d'instruccions és: LDD YZ+q, STD YZ+q.

Aquestes instruccions, a excepció de LDI, actualitzen el registre d'estat.

Format de les instruccions

El format de les instruccions de transferència de dades és el següent:

- Instruccions de transferència de dades directa.
 - o LDS Rd, K. Efectua la càrrega $Rd \leftarrow \text{mem}\langle K \rangle$

STS K, Rs. Efectua la guarda $\text{mem}\langle K \rangle \leftarrow R_s$, a on

Rd correspon a un registre

K és la posició de memòria de dades amb la que es treballa

- El codi màquina de la instrucció es compon de dues paraules, i té el format:

110p	---d	dddd	01--	----	KKKK	KKKK	KKKK
------	------	------	------	------	------	------	------

a on:

- p és 0 si és LDS i 1 si és STS
- ddddd és el registre
- K...K és la posició de memòria amb la que es treballa

- Instruccions de transferència de dades indirecta.

- LD Rd, XYZ. Efectua la càrrega $R_d \leftarrow \text{mem}\langle XYZ \rangle$

ST XYZ, Rs. Efectua la guarda $\text{mem}\langle XYZ \rangle \leftarrow R_s$, a on

Rd correspon a un registre

XYZ és el registre amb el que es treballa: X, Y o Z

Hi ha la possibilitat de pre-increment o pre-decrement del registre XYZ.

- El codi màquina de la instrucció és:

110p	ZYXd	dddd	00ab
------	------	------	------

a on:

- p és 0 si és LDS i 1 si és STS
- ddddd és el registre
- ZYX val 100, 010 o 001 segons s'empri Z, Y o X, respectivament
- ab és la codificació de la indirecció:
 - 0 → Indirecció normal (sense pre-decrement ni pre-increment)
 - 01 → preincrement
 - 11 → predecrement

- Instruccions de transferència de dades indirecta amb desplaçament

- LDD Rd, Y+q, LDD Rd, Z+q. Efectua la càrrega $R_d \leftarrow \text{mem}\langle XYZ+q \rangle$

STD Y+q, Rs, STD Z+q, Rs. Efectua la guarda $\text{mem}\langle XYZ+q \rangle \leftarrow R_s$, a on

Rd correspon a un registre

q és la constant de desplaçament

YZ és el registre Y o Z que s'empri en la instrucció

- El codi màquina de la instrucció és:

11(Y/Z)p	qqqd	dddd	qqqq
----------	------	------	------

a on:

- p és 0 si és LDS i 1 si és STS
- Y/Z val 1 quan s'empri Y, 0 quan s'empri Z.

- dddd és el registre.
- qqq qqqq és la constant de desplaçament, entre 0 i 127.

Exemples

- LDS R31, 0x842, que realitza la càrrega directa $R31 \leftarrow \text{mem}\langle 0x842 \rangle$ és codificat amb

1100 0001 1111 0100 | 0000 1000 0100 0010

- ST +Y, R20, que realitza l'operació

...primer s'incrementa Y: $Y \leftarrow Y+1$

...i després $\text{mem}\langle Y \rangle \leftarrow R20$

es codifica amb

1101 0101 0100 0001

- STD Z+65, R0, que realitza l'operació $\text{mem}\langle Z+65 \rangle \leftarrow R0$ es codifica amb

1101 1000 0000 0001

6.3.6 Instruccions de salt

Conjunt d'instruccions

Les instruccions de salt poden ser:

- **Incondicionals.** Efectuen un salt en l'ordre d'execució del programa sempre.

El conjunt d'instruccions està format per:

- o RJMP k. Efectua un salt relatiu a k posicions respecte a la posició actual del comptador de programes
 - o RCALL k. Guarda el contingut actual del comptador de programes en la pila i efectua un salt relatiu a una subrutina que es troba a k posicions respecte a la posició actual del comptador de programes.
 - o RET. Efectua el retorn de subrutina. Per fer-ho va a cercar l'adreça de retorn de la pila.
 - o RETI. És com RET però pel retorn de rutina de servei d'interrupció.
- **Incondicionals indirecte.** Són instruccions de salt incondicional en les que el desplaçament es troba en el registre Z. Està format per les instruccions:
 - o IJMP. Efectua un salt relatiu, amb desplaçament donat en el registre Z, respecte a la posició actual del comptador de programes
 - o ICALL. Guarda el contingut actual del comptador de programes en la pila i efectua un salt relatiu a una subrutina que es troba a Z posicions respecte a la posició actual del comptador de programes.
 - **Condicionals.** Efectuen un salt quan es compleix una condició determinada sobre algun bit del registre d'estat.

Segons el bit del registre d'estat estigui activat o no es té dues instruccions genèriques:

- o BRBS k. La instrucció ve de *Branch if Register Bit is Set*. És adir, saltar sobre bit activat.
- o BRBC k. La instrucció ve de *Branch if Register Bit is Clear*. És adir, saltar sobre bit no activat.

La condició de bit activat o no es realitza sobre els bits del registre d'estat: carreteig (C o *carry*), zero (Z), negatiu (N), excés (V o *overflow*) i l'interrupció).

Aquestes dues instruccions es divideixen en instruccions particulars que actuen sobre un bit concret:

- o BRCS k, BRCC k. Pregunten sobre si el bit de carreteig és 1 o no, respectivament.
- o BRLO k, BRSH k. Com en el cas anterior pregunten sobre si el bit de carreteig és 1 o no, respectivament.
- o BRZE (BREQ) k, BRNZ (BRNE) k. Pregunten sobre si el bit de zero és 1 o no, respectivament.
- o BRMI k, BRPL k. Pregunten sobre si el bit de nombre negatiu és 1 o no, respectivament.
- o BRVS k, BRVC k. Pregunten sobre si el bit d'excés és 1 o no, respectivament.
- o BRIE k, BRID k. Pregunten sobre si el bit d'interrupció està habilitat o no, respectivament.

Format de les instruccions

- El codi màquina de les **instruccions incondicionals** ve donat per:

010p	kkkk	kkkk	kkkk
------	------	------	------

- p val 0 per la instrucció RJMP i 1 per la instrucció RCALL.

- Per **instruccions incondicionals amb adreçament indirecte** el codi màquina de la instrucció ve donat per:

0110	zyx-	----	001p
------	------	------	------

- ZYX val 100, 010 o 001 segons s'empri Z, Y o X, respectivament
- p val 0 per la instrucció IJMP i 1 per la instrucció ICALL.

- Per instruccions de salt condicional el codi màquina de la instrucció és:

Per les instruccions de bit activat o BRBS és

1110	kkkk	kkkk	kbbb
------	------	------	------

Per les instruccions de bit no activat o BRBC és

1111	kkkk	kkkk	kbbb
------	------	------	------

a on:

- La codificació d'instrucció de salt ocupa els 4 bits més significatius
- K...k fa referència al salt relatiu, que va de +255 a 256 posicions de memòria
- bbb és la codificació del bit de salt, codificat en la forma següent:

0 → Condició sobre bit C

1 → Condició sobre bit Z

2 → Condició sobre bit N

3 → Condició sobre bit V

7 → Condició sobre bit I

Exemple

- BRNE 45 (que és equivalent a BRBC 1, 45). Realitza l'operació

Si el bit Z val 0 aleshores salta a PC+45, si no continua en seqüència

es codifica amb

1111 0001 0110 1001

- BRMI -45 (que és equivalent a BRBS 2, -45). Realitza l'operació

Si el bit Z val 0 aleshores salta a PC+45, si no continua en seqüència

es codifica amb

1110 1110 1001 1001

Observar que -45 = 111010011 quan es codifica en complement a la base sobre 9 bits.

6.3.7 Instruccions d'entrada/sortida

Són instruccions de treball amb la memòria d'entrada/sortida

Conjunt d'instruccions

Hi ha dues instruccions:

- IN Rd, PORT. Realitza l'operació $Rd \leftarrow \text{PORT}$.
Comunica l'exterior amb el processador a través d'un port d'entrada.
- OUT PORT, Rs. Realitza l'operació $\text{PORT} \leftarrow Rs$.
Comunica el processador amb l'exterior a través d'un port de sortida.

Format de les instruccions

- El codi màquina de les instruccions és:

0111	pAAd	dddd	AAAA
------	------	------	------

a on:

- P val 0 quan la instrucció és IN, i 1 quan és OUT
- ddddd és el registre de treball
- AA-AAAA és l'adreça del port d'entrada/sortida. Val entre 0 i 63.

Exemple

- OUT 32,R0, treu pel port de sortida 32 el valor de R0 i es codifica com
0111 1100 0000 0000
- IN 32, R0 guarda en R0 el valor del port d'entrada 32. Es codifica com
0111 0100 0000 0000

6.3.8 Instruccions de pila (*stack*) i de guarda d'estat

Quan es treballa amb la pila cal recordar dos detalls:

- L'operació sempre s'executa amb la posició de memòria a la que apunta l'apuntador de pila. En l'operació de PUSH primer es posa la dada i després s'incrementa l'apuntador de pila. Quan s'executa POP primer es decrementa l'apuntador i després es treu la dada.
- S'ha d'anar en compta en emprar la pila quan es treballa amb subrutines i interrupcions ja que s'empra la pila per guardar dades de retorn i dades temporals.

Conjunt d'instruccions

Hi ha dues instruccions que permeten treballar amb la pila:

- PUSH Rd. Posa el valor del registre Rd en la pila.
- POP Rs. Recupera el valor de la pila i el posa en el registre Rs.

Apart n'hi ha dues més que empen la pila per guardar i recuperar l'estat (compta que només fan referència als bits V, N, Z i C i no al bit I d'establiment d'interrupció):

- SAVE. Posa el valor del registre d'estat en la pila.
- RESTORE. Recupera el valor de la pila i el retorna en el registre d'estat.

Format de les instruccions

- El codi màquina de la instrucció és:

0110	e00d	dddd	011p
------	------	------	------

a on:

- ddddd és el registre de treball
- p val 1 per posar en pila (PUSH i SAVE) i 0 per treure de pila (POP i RESTORE).
- e val 1 en les instruccions SAVE i RESTORE

Exemple

- PUSH R0 es codifica amb
0110 0000 0000 0111
- POP R0 es codifica amb
0110 0000 0000 0110
- SAVE es codifica amb
0110 1000 0000 0111
- RESTORE es codifica amb
0110 1000 0000 0110

6.3.9 No fer res

Finalment queda la instrucció de no fer res o NOP.

S'empra per fer rutines d'espera.

Format de la instrucció

- El codi màquina de la instrucció és:

0000	0000	0000	0000
------	------	------	------

Exemple

NOP

6.4 Resum del capítol

El capítol ha introduït el repertori d'instruccions que s'executen en EduP12 i ha analitzat en detall els diferents tipus d'instruccions que el configuren, especificant en cada instrucció el conjunt de camps que la configuren.

Associat amb cada instrucció que treballa amb dades hi ha l'adreçament. S'ha vist que EduP12 admet un conjunt important de modes d'adreçament i s'han detallat en cada instrucció.

Els apèndixs del llibre complementen la informació donada del processador EduP12 en els capítols 5 i 6.

- La taula de l'apèndix A1 condensa el repertori d'instruccions complert del processador EduP12, indicant el nombre de cicles de rellotge que necessita cada instrucció, la codificació, el nombre de paraules que requereix i els bits del registre d'estat que queden afectats per la seva execució
- L'apèndix A2 és la guia d'usuari de les instruccions. Desglossa per totes les d'instrucció la transferència de dades que s'estableix en la CPU quan s'executa la instrucció.