

Accessi alla memoria

Region	Bufferable	Cacheable	Executable
Code	yes	write through	yes
SRAM	yes	write back	yes
Peripherals	no	no	no
External RAM	no	write back	yes
External device	no	no	no
System	no	no	no

E' possibile eseguire del codice presente sulla RAM, spostando il PC alla porzione della RAM.

E' possibile anche modificare il codice(cacheable write through)

Load and store

Load	Store	Size and type
LDR	STR	word (32 bits)
LDRB	STRB	byte (8 bits)
LDRH	STRH	halfword (16 bits)
LDRSB	–	signed byte
LDRSH	–	signed halfword
LDRD	STRD	two words
LDM	STM	multiple words

LDM e **STM** permettono di copiare più parole assieme.

Viene copiata una parola per ciclo! Non vengono caricate massivamente tutte nello stesso

ciclo.

LDRSB e **LDRSH** permettono di completare il resto della parola con il bit di segno.

Le altre istruzioni sono auto esplicative.

Addressing

Gli offset per il pre-indexing e post-indexing possono essere:

- costanti da 12 bit
- registri
i quali possono essere shiftati a sinistra di 3 posizioni.

Pre-indexing

Permette di aggiungere un offset al registro da cui caricare l'indirizzo **prima** del caricamento.

LDR r3, [r0, #4] carica in **R3** il valore contenuto nell'indirizzo **R0+4**

Writeback

Permette di aggiornare il registro con l'offset dopo il caricamento

LDR r3, [r0, #4]!

Stessa cosa di sopra ma aggiorna anche **R0** con **R0=R0+4**

Post-indexing

Permette di aggiornare un registro con un offset **dopo** il caricamento della memoria

LDR r3,[R0],#4 carica in **R3** il valore contenuto in **R0** e **poi** aggiorna **R0** con **R0=R0+4**.

Manca il ! in quanto c'è **sempre** il **write-back**

Look-up table

Sono tipo delle *literal pool* ma create dal programmatore e non dall'assemblatore.

```
AREA    |.text|, CODE, READONLY
Reset_Handler PROC
    EXPORT Reset_Handler [WEAK]
    MOV r2, #8    ;after some calculus
    LDR r0, =lookup
    LDRB r4, [r0, r2]
stop B stop          ;stop program
lookup DCB 1, 4, 9, 16, 25, 36, 49,
64, 81, 100
    ENDP
```