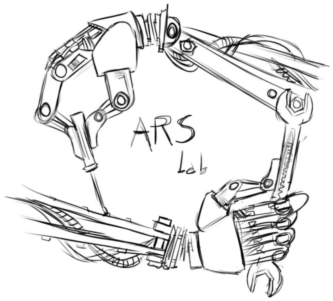
Lo Stack

Corrado Santoro

Dipartimento di Matematica e Informatica
santoro@dmi.unict.it



Corso di Architettura degli Elaboratori

Cos'è lo Stack (Pila)

- Lo **stack** è una struttura dati (insieme di dati) di tipo *LIFO* (Last In, First Out)
- Utilizzato dal microprocessore per:
  - Salvare indirizzi di ritorno
  - Memorizzare registri temporaneamente
  - Gestire chiamate di funzione e interrupt
- Risiede in una parte della RAM
- Utilizza un registro speciale, lo **Stack Pointer – SP**, che punta alla cima dello stack (top-of-the-stack)

Operazioni Fondamentali

PUSH *val*

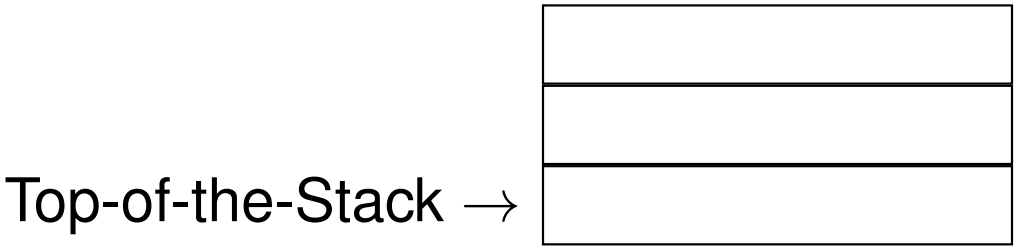
- Inserisce il dato *val* sulla “cima” dello stack

POP *var*

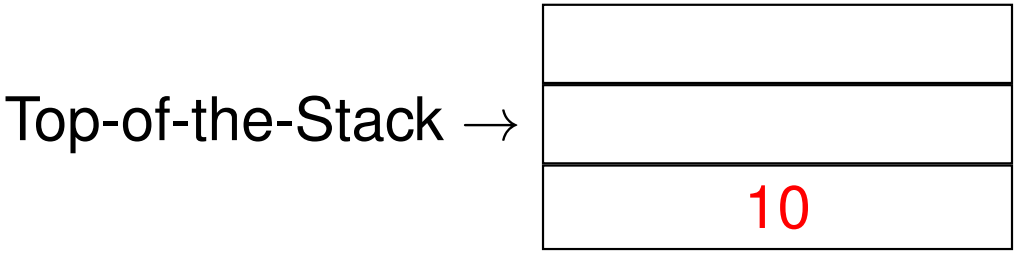
- Preleva (e rimuove) l’ultimo dato inserito nello stack e lo assegna alla variabile *var*

Esempio Grafico dello Stack

- Stack “vuoto” di dimensione 3 elementi



- PUSH 10



Esempio Grafico dello Stack

- PUSH 20

Top-of-the-Stack →

- PUSH 30

Top-of-the-Stack →

Esempio Grafico dello Stack

- PUSH 40
  ERRORE!! Non c'è più spazio nello stack!!

Top-of-the-Stack →

30
20
10

- POP *var*

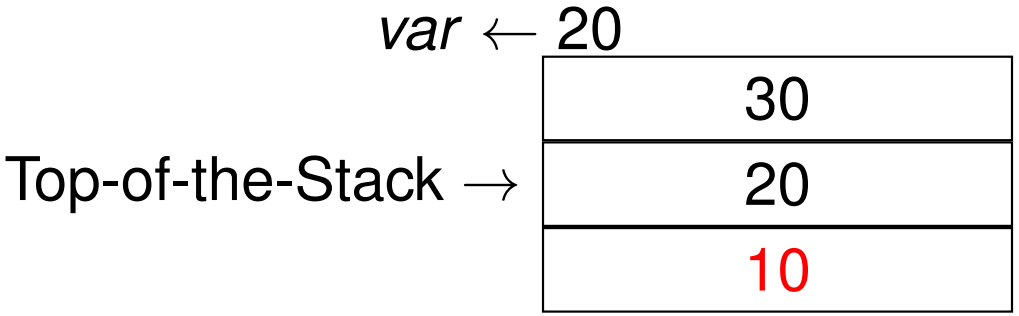
var ← 30

Top-of-the-Stack →

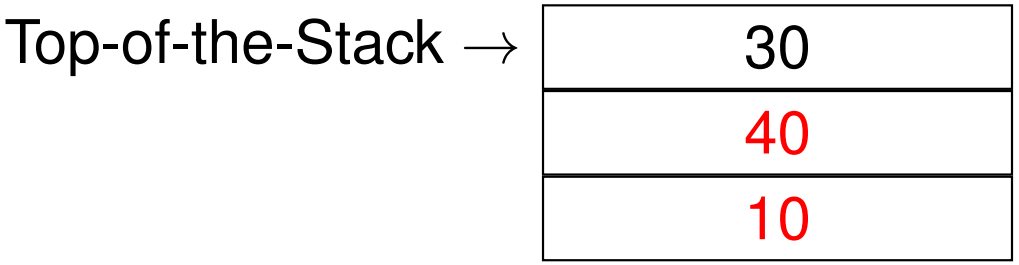
30
20
10

Esempio Grafico dello Stack

- POP *var*

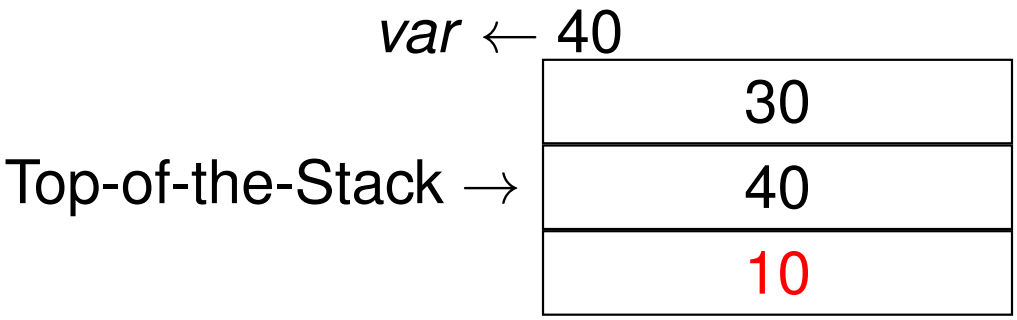


- PUSH 40

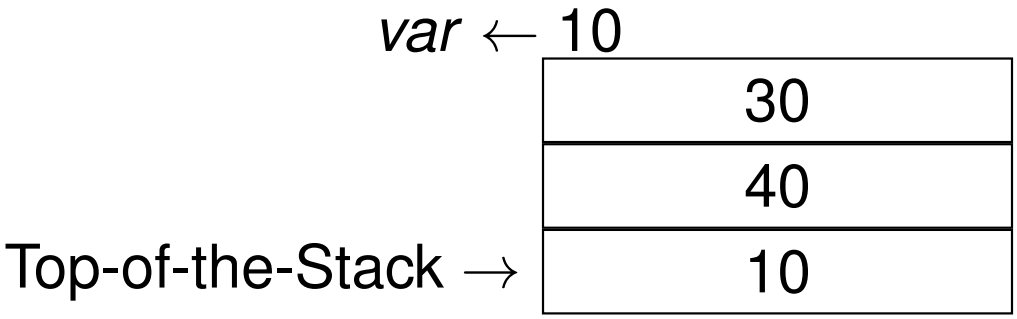


Esempio Grafico dello Stack

- POP *var*



- POP *var*



Lo Stack Pointer (SP)

- E' un registro speciale presente in tutte le CPU che punta ad un indirizzo di memeoria che costituisce la cima dello stack
- Nel processore ARM è il registro R13
- Può crescere:
  - Verso indirizzi di memoria **decrescenti** (comune)
  - Verso indirizzi **crescenti**
- Ogni operazione di stack modifica automaticamente lo SP
- Le operazioni PUSH e POP utilizzano registri come argomenti

Operazioni Fondamentali ed Equivalenza

PUSH Rx

```
1 STR Rx, [R13]      ; store
2 SUB R13, R13, #4   ; decrement after store
```

POP Rx

```
1 ADD R13, R13, #4   ; increment before load
2 LDR Rx, [R13]      ; load
```

In ogni caso è possibile sia la dicitura R13 che SP

PUSH/POP su ARM ed equivalenza con STM/LDM

Il processore ARM non ha istruzione esplicite di PUSH/POP ma è possibile utilizzare, in modo equivalente, le istruzione di Load/Store Multiple Registers

PUSH R*x*

```
STMDA R13!,{Rx} ; store Rx on [R13] and decrement after
```

POP R*x*

```
LDMIB R13!,{Rx} ; increment before loading Rx from [R13]
```

Stack e Chiamate di Funzione

Chiamate di Funzione in Assembly

- Lo stack è **fondamentale** per le chiamate di funzione
- Le chiamate di funzione in Assembly si effettuano utilizzando l'istruzione `CALL addr`
- `addr` è 'indirizzo della locazione di memoria dove è presente il **codice della funzione** invocata
- Al termine della funzione, l'esecuzione del programma **deve riprendere** dall'istruzione **successiva alla CALL**

Stack e Chiamate di Funzione

- Nel momento in cui viene eseguita una **`CALL addr`**, il valore del Program Counter viene "spinto" (push) nello stack:

```
0x1234:  ...
0x1238:  CALL 0x2000
0x123C: ...
```

- In questo caso, la CALL inserisce **`0x123C`** nello stack ed effettua un branch alla locazione `0x2000`

- La terminazione della funzione deve essere effettuata usando l'istruzione **`RET`** (*return*)

```
0x2000:  ...
0x....:  ...
0x....:  RET
```

- Questa istruzione estrae (pop) una word dallo stack e la assegna al Program Counter, in modo che l'esecuzione possa riprendere dall'**istruzione successiva** alla CALL

Chiamata di Funzioni in ARM

Branch-and-Link

- Su ARM **non esiste** un'istruzione CALL esplicita
- Esiste un meccanismo simile offerto dall'istruzione **"Branch-and-Link" – BL *addr***
- Questa istruzione **copia il PC** sul **Link Register, R14** ed effettua il branch all'indirizzo specificato
- Per effettuare il "ritorno da funzione", è sufficiente ricopiare il valore del Link Register sul Program Counter:

```
MOV PC, LR
```

Chiamata di Funzioni in ARM

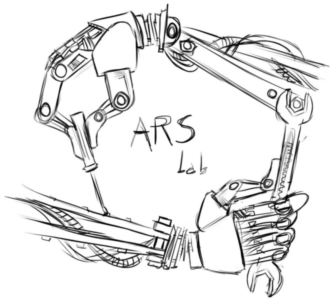
Branch-and-Link

```
0x1234:   ...
0x1238:   BL 0x2000
0x123C: ...
......:   ...
0x2000:   ...
0x....:   ...
0x....:   MOV PC, LR
```

Lo Stack

Corrado Santoro

Dipartimento di Matematica e Informatica
santoro@dmi.unict.it



Corso di Architettura degli Elaboratori