

## Esercizio: Dipendenze

**Dipendenza dai dati** : per la quale l'istruzione  $j$  dipende dall'istruzione  $i$  se  $i$  produce, direttamente o transitivamente (ossia tramite una o più istruzioni intermedie) un risultato richiesto da  $j$ .

**Dipendenza dal controllo** : la quale determina l'ordinamento di una istruzione rispetto ad un salto condizionale, così che essa esegua solo quando dovuto rispetto all'esecuzione del salto.

**Dipendenza dai nomi** : la quale ha luogo allorché due istruzioni, tra le quali non vi sia flusso di dati, usano lo stesso registro o la stessa locazione di memoria.

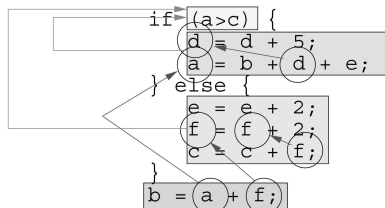
Si illustrino tutte le dipendenze presenti nel seguente frammento di programma in linguaggio C, assumendo che il programma non faccia riferimento ad altri dati, che tutti i valori siano definiti prima dell'uso, e che soltanto  $b$  e  $c$  siano usati successivamente alle istruzioni date:

```

if (a>c) {
    d = d + 5;
    a = b + d + e; }
else {
    e = e + 2;
    f = f + 2;
    c = c + f; }
b = a + f;
    
```

## Soluzione

A  
B | 1  
  | 2  
C | 1  
  | 2  
  | 3  
D



legenda:



dipendenza dal controllo: X dipende da Y



dipendenza dai dati: X dipende da Y

dipendenze		
dal controllo	dai dati	dai nomi
B → A	B.2 → B.1	
C → A	C.3 → C.2	
	D → C.2, B.1 (via B.2)	

## Esercizio Microprogrammazione

Sia data una unità di traduzione degli indirizzi di pagina logica in pagina fisica, che riceve in ingresso un indirizzo di pagina logica, e restituisce il corrispondente indirizzo fisico (allineato alla mezza parola, cioè 2B) se la pagina è presente nella tabella di rilocazione memorizzata all'interno dell'unità, o un segnale di page fault altrimenti.

Si assuma che:

- la tabella di rilocazione sia memorizzata in un banco di registri a 32 bit e che la presenza della pagina logica in memoria fisica sia indicata dal valore del bit più significativo (bit 31) di ogni entry. Inoltre il corrispondente indirizzo fisico sia codificato nei bit 0-30 di ogni entry.
- prima di essere emesso in output l'indirizzo fisico deve essere traslato a sinistra di 1 bit (allineamento mezza parola).
- l'unità operi secondo la seguente specifica:

```
Repeat
begin
  OUT.ok := 0;           // indica che non c'è un dato significativo in output
  if (IN.ok = 1) then    // se IN.ok = 1 c'è un indirizzo logico da tradurre sulla linea di ingresso
  begin                // l'indirizzo logico è rappresentato come un intero (da 0 a N-1) ed è presente nella linea di ingresso IN.dato
    B := TABELLA[IN.dato]; // TABELLA[] contiene la tabella di rilocazione
    if (B31 = 1) then    // bit 31 = 1 indica la presenza di pagina logica in memoria fisica
    begin              // OUT.dato rappresenta la linea di output
      OUT.dato := shift_sx(B); // shift_sx() esegue la traslazione a sinistra di 1 bit
      OUT.ok := 1;           // indica che c'è un dato significativo in output
    end;
    else FAULT := 1;       // se la pagina non è presente in memoria fisica si segnala un page fault
  end;
end
```

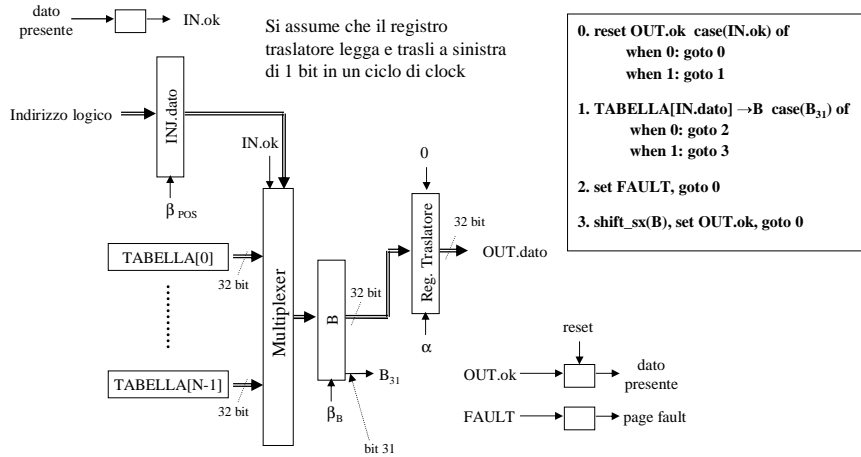
Si descriva la PO necessaria per implementarla e un microprogramma in linguaggio TS per la parte controllo corrispondente.

## Livello Firmware

### ▪ Microlinguaggio TS

```
etichetta : microoperazione
  case condizione logica
  when valore 1 => indirizzo successivo 1
  when valore 2 => indirizzo successivo 2
  ...
  when valore n => indirizzo successivo n
  end case
```

## Una soluzione: PO e $\mu$ -programma



Esercizi

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 41