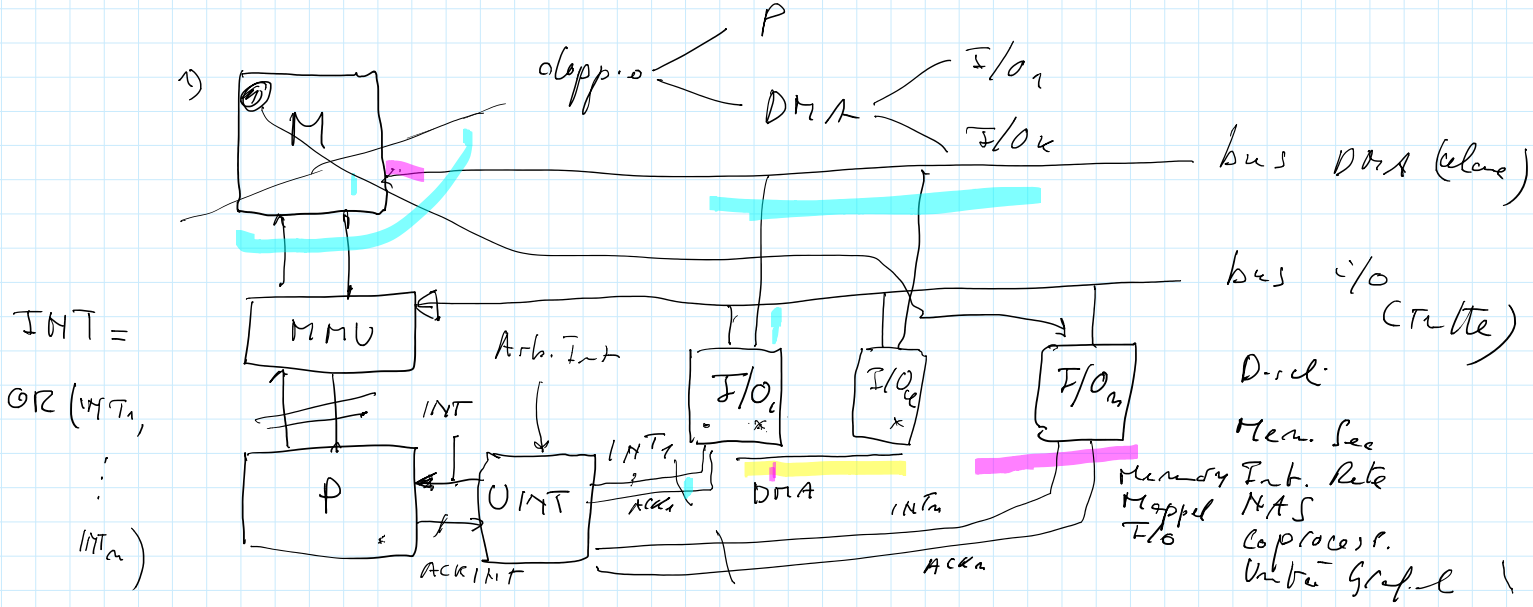


I/O ≡ Ingresso / uscita, interruzione ...



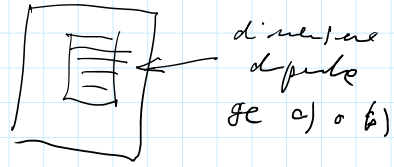
INT =  
OR (INT<sub>1</sub>,  
:  
INT<sub>n</sub>)

1) Processore LD/ST per ingresso/uscita

2) I/O = registri ← programma

a) memoria locale

b) memoria collegata DMA



Op. Ingresso

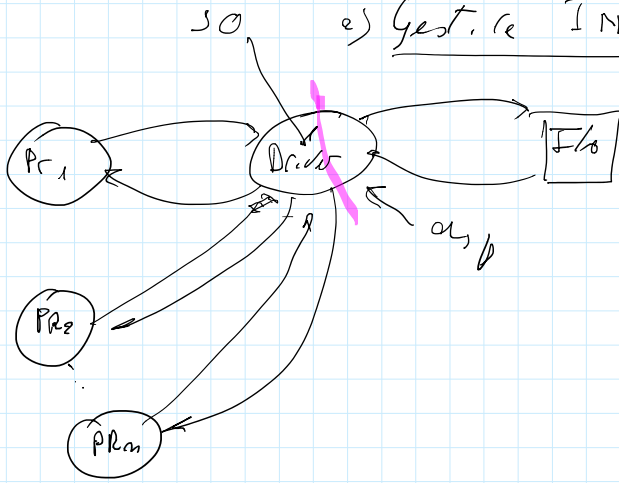
1) processore applica i registri con LD/ST

2) dispositivo locale

Driver → Modul. SO

1) Virtualizzare disp = Arbitraggio

2) Gest. (e) INTERAZIONE ed INTERRUZIONI



repeat

< collect r. data >

< sample r. data >

repeat

< r. data disp >

< r-sp. disp >

util

programme

driver frame

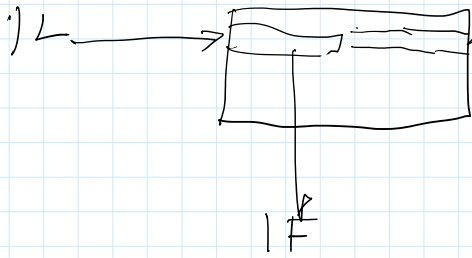
in alt. ut. l

≡

interazione ut. l

disp.  
low

I/O o Memorie Principali



I/O

Istr. I/O  
LM

Mem. Mapped

proprietà

a) virtuale per prog.

a) L3/S7

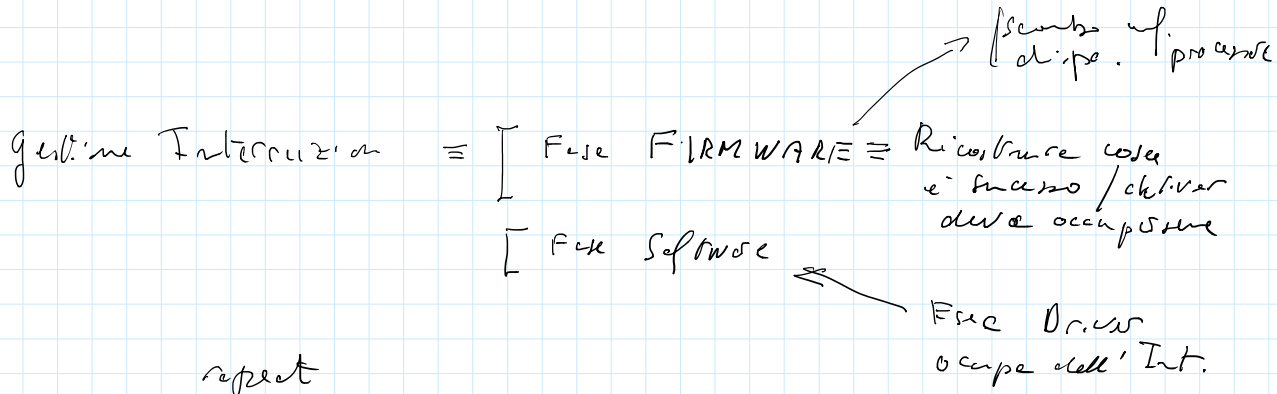
b) bus I/O collegato al processore

b) bus I/O collegato alla MMU

virtuale

DATA

~~Mem. Locale~~



- VINT  $\equiv$  repeat
- 1) almeno un INT<sub>i</sub> = 1
  - 2) INT = 1 per il processore
  - 3) attendi Ack del processore  $\leftarrow$
  - 4) invia Ack ad uno dei disp. t.v.:  $\leftarrow$  arbitrio.
- while
- II  
 arbitro  
 - centralizzato  
 - gerarchico

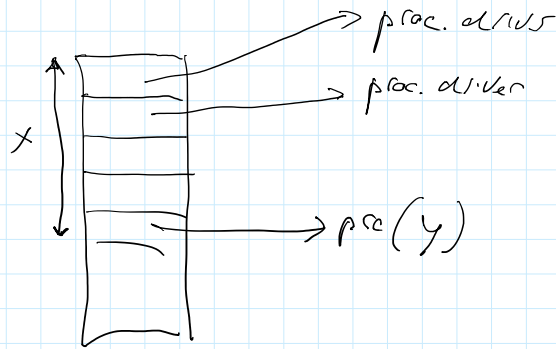
Accettare int, processare attuale info del dispositivo

Disp  $\rightarrow$  P

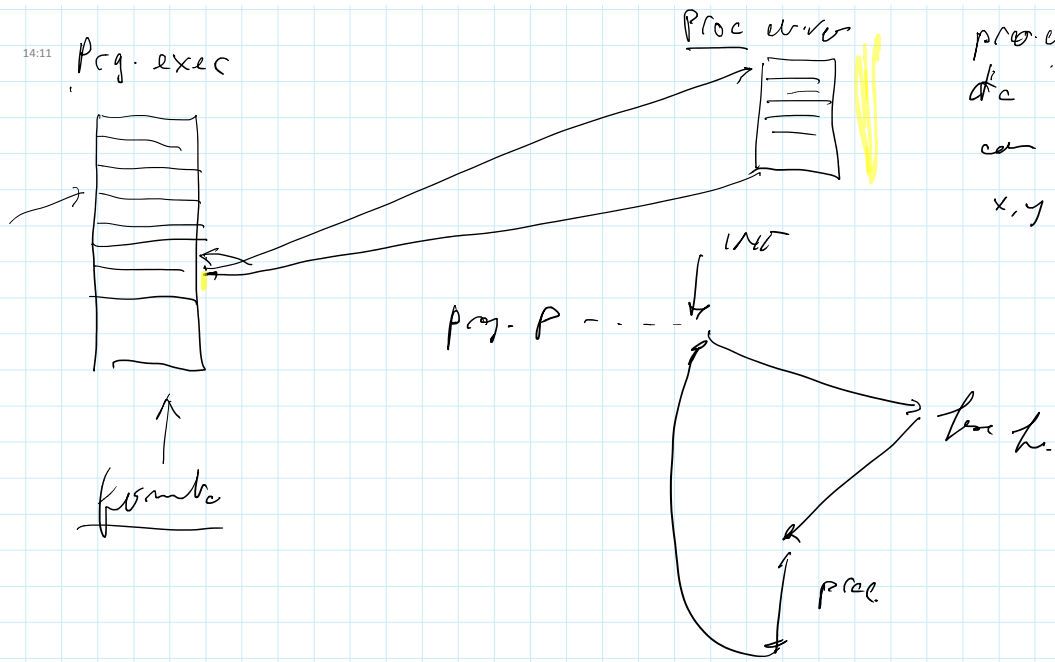
- a) identif. disp  $x$
- b) identificare regione int.  $y$

vettore di interruzione

Tabelle punto  
memoria principale  
SO



Fase firmware  
 $\equiv \rightarrow (int, x, y)$   
 Trasl. int in  
 una ind. corrente  
 di processo



procedura pubblica  
da tabella(x)  
con parametro y  
x,y = vett. ut.

# TRATT. INTER.

R60 = procedura  
 selto all' ind. tabella } INT  
 R61 = per } PROC  
 R62 = per }  
 R63 = ind. direzione }

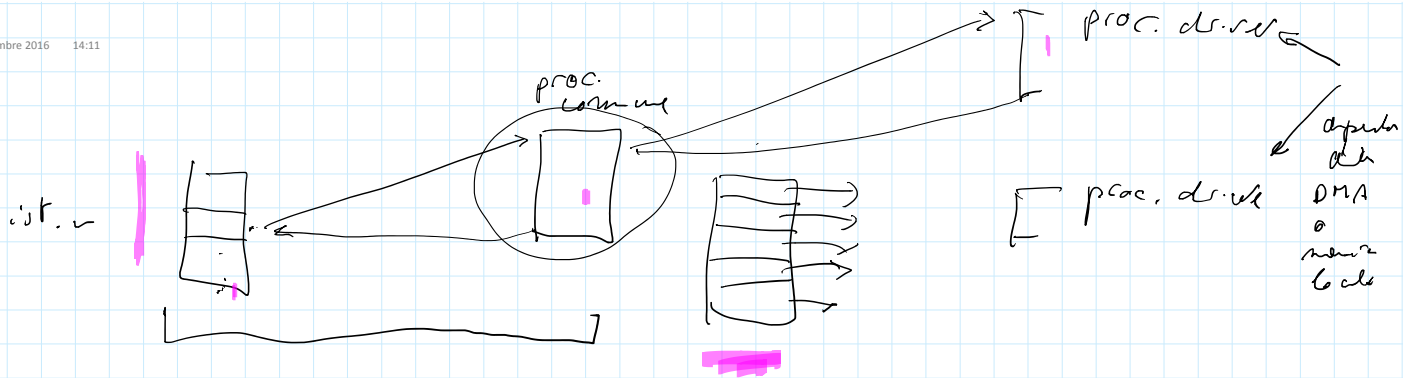
tratt.int : reset (INT), set (ACK INT), int<sub>1</sub> ← selto UNIT

int<sub>1</sub> : case RDY IN, OR (ESITO) of  
 φ , - : nop, int<sub>1</sub>  
 1 , 1 : ESITO → ESITO1  
 reset (RDY IN), tratt. ecc.  
 1 , φ : reset (RDY IN),  
DATA IN → RG [61], int<sub>2</sub>

int<sub>2</sub> : case RDY IN, OR (ESITO) of  
 φ , - : nop, int<sub>2</sub>  
 1 , 1 : ESITO → ESITO1, reset (RDY IN),  
 tratt. ecc.  
 1 , φ : reset (RDY IN), DATA IN → RG [60]  
 IC → RG [63], RG [60] → IC, ch φ

selto  
 pero di ch φ





funzione di read in un  
 altro a questo modo  
 EG [60] + DATA 17

utilizzo  
 di DMA/memoria  
 locale

# Compito a casa

```
for (i = 0, i < n, i++)  
  { for (j = 0, j < n, j++)  
    { x[j] = x[j] + x[j]  
      sv[i] = sv[i] + x[j]  
    }  
  }
```

1) working set  
= pagine su  
cui prog. lavora

2) n° fault codice

3) n° fault dati

4) il n° fault dati  
dipende dal  
n° di pagine  
nel codice per  
la singola struttura  
dati  
(STRUTTURA  
DATI)