

CPU

Introduzione

e formato da

unità di elaborazione (data-path)

registri

ALU

...

unità di controllo (UC)

genera i segnali di controllo per il data-path

Unità di controllo

può essere modellata come un **circuito sequenziale** che riceve **segnali dall'esterno**

C'_{in}

e dall'**unità di di elaborazione** C''_{in}

sulla base dei segnali e dallo stato corrente essa produce i **segnali di controllo per l'esterno**

C'_{out}

e per l'**unità di elaborazione** C''_{out}

ad ogni "botta" di clock

genera segnali per eseguire "microistruzioni"

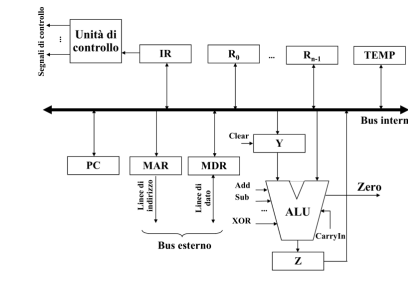
Operazioni elementari

FETCH

DECODE

OPERATION

WRITING



Architettura

Trasferimento tra registri

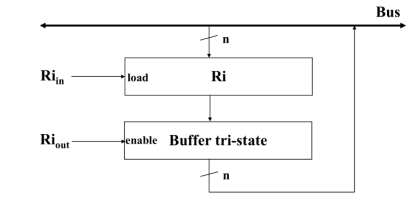
ogni registro R_i possiede un **interfacci** controllata da 2 segnali

Ri_{in}

carica nel registro il valore presente nel bus

Ri_{out}

carica sul bus il valore presente nel registro



Come avviene un trasferimento tra registri

Attivare $R1_{out}$

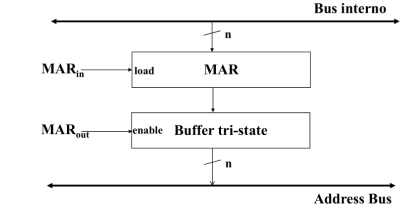
Attivare $R2_{in}$

Connessione MAR

Memory Address Register

un registro che fa da tramite tra il **bus interno** e l'**Address BUS**

funzionamento



Segnali interni

MAR_{in}

legge i valori dal **bus interno**

MAR_{in}

scrive i valori sul **ABUS**

ACCESSO in meoria

dice all' registro di scrivere l'indirizzo sul bus

Abilitandolo tramite Ri_{out}

esso viene memorizzato nell' MAR

Abilitandolo tramite MAR_{in}

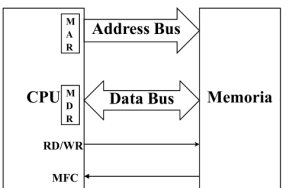
in un altro istante viene scritto sul ABUS da un Buffer tri-state

Abilitandolo tramite MAR_{out}

$MAR \leftarrow Ri$

Quindi si esegue la seguente micro istruzione

Interfaccia CPU-Memoria



dopo aver ricevuto i segnali di lettura/scrittura

la memoria impiega un po' di tempo a completare un operazione

una volta completata l'operazione

la meoria un segnale che indica la fine dello operazione

MFC

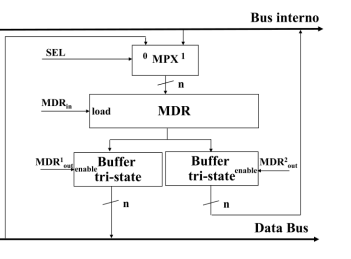
Problemi

Temporizzazione

ogni segnale deve rimanere attivo fino alla fine dell' istruzione

tendo conto dei ritardi dei vari componenti

Connessione MAR



il registro che fa da tramite tra il bus interno e il **DATA BUS**

Memory Data Register

Segnali intetrni

0:DataBUS

Gestisce dove deve essere preso il valore da inserire sul MDR

SEL

1:BUS_INT

Viene salvato il dato preso dal DataBUS/BUS_INT

MDR_{in}

Scrive il valore del MDR sul DBUS

MDR^1_{out}

Scrive il valore del MDR sul BUS_INT

MDR^2_{out}

Scrittura in memoria

predene i valori dal bus interno

Attivando MDR_{in}

si salva il valore sul MDR

Attivando MDR^1_{out}

scrive sul databus

accesso in memoria

prendere i valori dal DataBus

Attivando MDR_{in}

si salva il valore sul MDR

Attivando MDR^2_{out}

scrive sul Bus interno

Attivando Ri_{in}

Si salva il valore in un registro

mettendo $SEL=1$

viene scelto da dove viene preso l'input per MDR

mettendo $SEL=0$

viene scelto da dove viene preso l'input per MDR