ESECUZIONE ISTRUZIONE

giovedì 10 novembre 2022 13:36

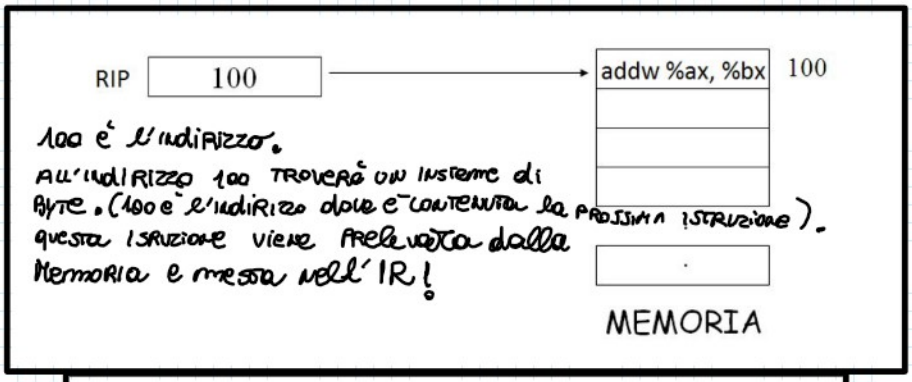
L'IP DENTRO HA UNA SEQUENZA DI BIT! QUESTA SEQUENZA DI BIT VIENE INTERPRETATA COME INDIRIZZO.

↳ INDIRIZZO DI MEMORIA DOVE È CONTENUTA LA PROSSIMA ISTRUZIONE DA ESEGUIRE.

Program Counter (PC), ovvero il contatore delle istruzioni, contiene l'indirizzo della prossima operazione da eseguire.

Instruction Register (IR), il registro delle istruzioni, contiene l'istruzione da eseguire.

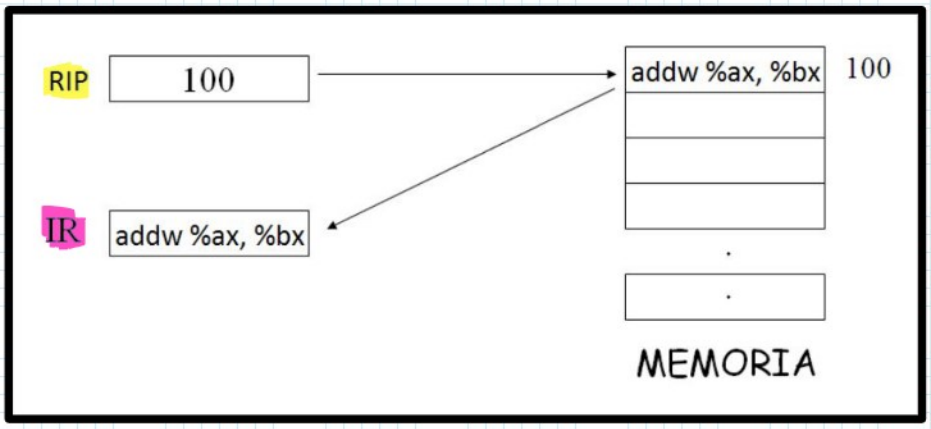
1.



100 È L'INDIRIZZO. ALL'INDIRIZZO 100 TROVERÒ UN INSIEME DI BYTE. (100 È L'INDIRIZZO DOVE È CONTENUTA LA PROSSIMA ISTRUZIONE). QUESTA ISTRUZIONE VIENE PRELEVATA DALLA MEMORIA E MESSA NELL'IR!

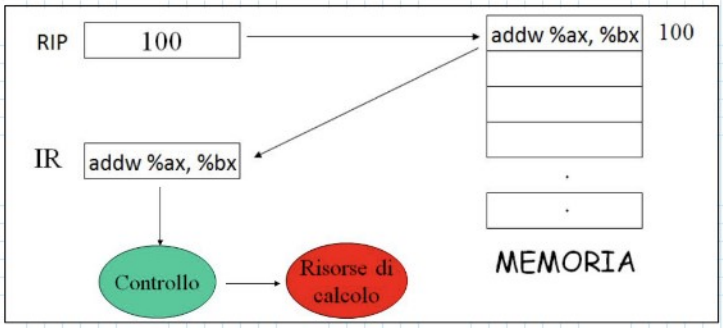
Durante la fase di fetch, l'istruzione viene copiata dalla memoria di lavoro nell'IR

2.



DA ORA IN POI, TUTTO IL PROCESSORE PUÒ LEGGERE LA CODIFICA BINARIA DELLA MIA ISTRUZIONE MACCHINA, DALL'IR.

Nella fase di decode, in funzione dei bit che la CU leggerà dall'IR, determinerà quali sono le risorse di calcolo che devono essere utilizzate per andare ad implementare la semantica di quell'istruzione.

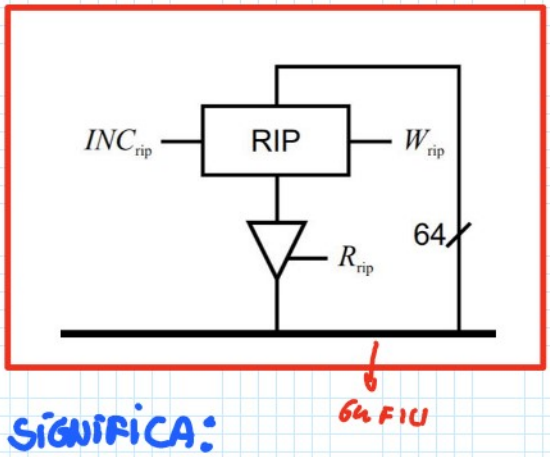


Ci sarà un segnale di controllo che ABILITERÀ la SCRITTURA sul registro destinazione. (Lo vedremo)

L'IP VIENE INCREMENTATO IN CIASCUNA FASE DI FETCH.

21:13

COME SI FA?



INCREMENTO DEL RIP CON SOMMATORE DEDICATO.
INCREMENTIAMO DI UNA CERTA QUANTITÀ DI BYTE LA QUANTITÀ DI RIP.

Potrei dire: porto il valore di IP in un registro 1, il valore della quantità di byte da sommare nel registro 2, fare la somma con l'alu, e il risultato metterlo in IP. E va benissimo ✓.
Il problema è che è un'operazione lenta! perché ho due registri temporanei davanti alla mia ALU.

SIGNIFICA:

1. COPIAMO RIP IN UNO DEI REGISTRI TEMPORANEI;
2. SCRIVERE NELL'ALTRO REGISTRO TEMPORANEO L'INCREMENTO;
3. IL RISULTATO VIENE RISCRITTO DENTRO RIP;

PER INCREMENTARE IL VALORE DEL PC HO NECESSITÀ DI 3 PASSI:
PERCIÒ QUESTI 3 PASSI LI DOVREI SVOLGERE PER L'ESECUZIONE DI OGNI ISTRUZIONE!

NON VA BENE, SI USA INVECE UN ADDER DEDICATO!

(INTERNO al RIP)

Realizziamo il registro RIP come un registro a INCREMENTO! È un registro che ha i suoi flip flop che mantengono i dati, E CONTIENE al SUO INTERNO UN sommatore dedicato.

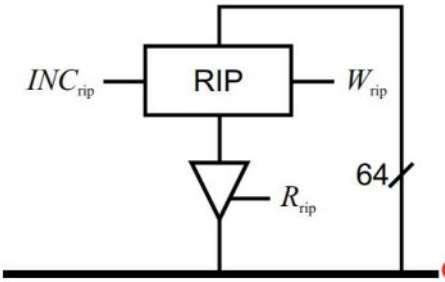
ALZANDO UN segnale di CONTROLLO, questo sommatore dedicato mi permetterà di INCREMENTARE RIP.

CHIARAMENTE devo AVERE anche UN BUFFER THREE-STATE IN USCITA perché devo SCRIVERE per aggiornare sul BUS di sistema condiviso;

ORA, CHE CI faccio con l'INCREMENTO di RIP?

UNA VOLTA CHE IO SO IL VALORE DI RIP LO DEVO TRASFERIRE ALLA MEMORIA, CON LO SCOPO POI DI ANDARCI A PRELEVARE IL CODICE MACCHINA DELL'ISTRUZIONE DA METTERE IN IR.
VADO SULLA MEMORIA TRAMITE IL BUS di SISTEMA. (VERSO la memoria)

IO ORA IMMETTO IL VALORE DI RIP nel DATA BUS.



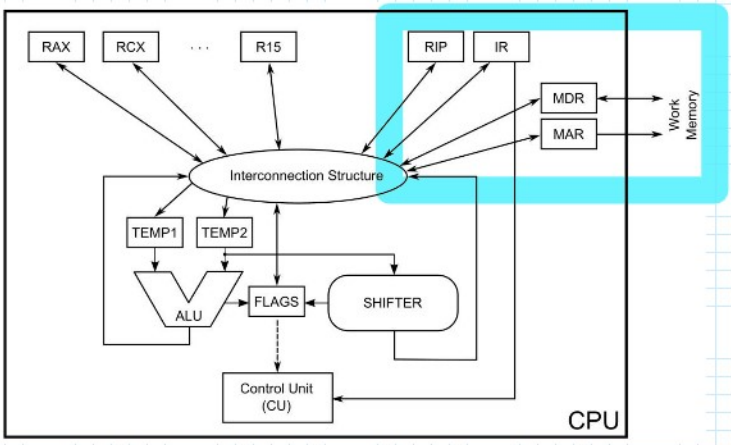
Dopo che ho SCRITTO qui IL valore di RIP deve CONNETTERSI con IL BUS di SISTEMA che va VERSO la MEMORIA.

Devo connettere questi 64 fili, con i 64 fili che escono dal PROCESSORE e vanno VERSO la memoria.

BUS di SISTEMA.

↳ DATA BUS della CPU INTERNO!

NON È CHE POSSO ATTACCARE I fili.

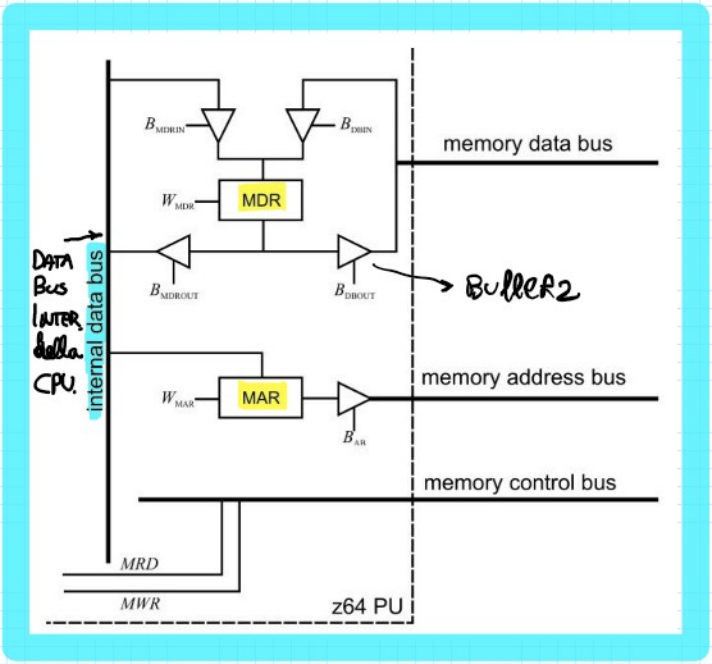


← GIUSTO PER CAPIRE: SIAMO IN QUESTA ZONA.

IL BUS di SISTEMA RIGUARDA anche i dati, le ISTRUZIONI che TRASFERISCO da e VERSO la memoria.

DEVO SVILUPPARE un CIRCUITO che mi CONSENTA di COLLEGARE IL DATA BUS COL BUS DI SISTEMA.
IL DATA BUS È UNO, IL BUS di SISTEMA È COMPOSTO da BUS INDIRIZZI e DATI.
SICCOME IO DEVO TRASFERIRE alla memoria un INDIRIZZO (lettura) e UN dato (SCRITTURA)
DEVO TROVARE un modo di DISACCOPPIARE IL DATA BUS INTERNO al PROCESSORE CON 2 BUS DIFFERENTI che vanno VERSO la MEMORIA.

IN PIÙ la memoria è LENTA, PERCIÒ DEVO MANTENERE L'INFORMAZIONE STABILE PER TUTTA la DURATA dell'OPERAZIONE.



ABBIAMO DUE REGISTRI: MAR & MDR;

MAR = Memory Address Register
MDR = Memory Data Register

Perché abbiamo un BUS di indirizzi e un BUS dati;
IL DATO VA SCRITTO NELL' MDR.
L'INDIRIZZO NELL' MAR.
I DUE REGISTRI SONO A 64 bit.

ALLORA: NEL DATA BUS INTERNO DELLA CPU, TRANSITA IL VALORE DEL RIP, E DEVO ANDARE IN MEMORIA PER PRELEVARE L'ISTRUZIONE SUCCESSIVA. Dopo aver abilitato il buffer three-state per mettere sul databus interno il contenuto dell' ISTRUCTION POINTER, abiliterò il segnale di scrittura sul **MAR**. A questo punto ho una copia del RIP in MAR, a questo punto l'indirizzo dell'istruzione successiva è conservata all'interno di MAR e poi devo fare fluire il contenuto di questo registro verso la memoria abilitando il buffer three-state!
Dopo tot di tempo scriverà sul data BUS il dato del codice macchina e arriveranno questi bit sul memory data BUS. Devo farne una copia per poi scriverlo nell' IR.
Abilitando il Buffer-three state, abilito la scrittura su MDR.
E trasferisco il contenuto di MDR sul BUS interno, e così LA POSSO SCRIVERE all' interno di IR.

SCRIVO IL VALORE di una VARIABILE CONTENUTA IN MEMORIA:

Posso scrivere qualsiasi dato proveniente dall'internal data Bus in MDR.
questo significa trasferire un dato dal registro alla memoria;

SCRIVERÒ IL DATO SU MDR PARTENDO DALL' INTERNAL DATA BUS PER MANDARLO sul memory DATA BUS, SCRIVO IL MIO DATO SULL'INTERNAL data BUS, e questo dato può provenire da un registro di uso generale e lo scrivo su MDR ... e lo mando verso la memoria!

MDR È UN REGISTRO BIDIREZIONALE CHE FUNZIONA DA APPOGGIO PER TUTTI I TRASFERIMENTI DI MEMORIA.

Come fa la memoria a capire se deve leggere dati dal MEMORY DATA BUS O SCRIVERE DATI SUL MEMORY DATA BUS?
GLIELO DICE IL PROCESSORE SFRUTTANDO IL BUS di CONTROLLO!
Sul Memory control Bus devo prevedere almeno due segnali di controllo: **MRD, MWR**.
Memory Read e Memory write!

Se voglio SCRIVERE sulla memoria, scrivo il mio indirizzo sul MAR e abilito il Buffer three state, mando l'indirizzo alla memoria.
Dopo aver scritto il dato che voglio scrivere in memoria, in MDR abilito anche il Buffer 2, così la memoria riceve anche il dato che deve scrivere, a quel punto abilito il segnale MWR chiedo alla memoria di prendere i dati che viaggiano sul memory data BUS e scriverli a quell'indirizzo su MAR.

Se voglio leggere dalla memoria, metto un indirizzo sul memory address bus, abilito MRD, a quel punto la mia memoria scriverà i dati sul memory data bus, abilitando il Buffer three state e col segnale di write, faccio una copia interna al Processore;

LA CU CHE GESTISCE IL TUTTO!