

ARCHITETTURA DEGLI ELABORATORI

RIEPILOGO

RAPPRESENTANTE: Scelta della grandezza fisica (Es. una tensione elettrica fra due punti di un circuito);

RAPPRESENTATO: Identificazione dell'informazione che si vuole elaborare (Es. Il valore numerico che esprime la temperatura di un locale in gradi centigradi);

Tipi di rappresentazione:

1) **RAPPRESENTAZIONE ANALOGICA:**

- Vantaggi: rappresentazione fedele ed intuitiva;
- Svantaggi: è una rappresentazione vulnerabile che aumenta gli errori di elaborazione;

2) **RAPPRESENTAZIONE DIGITALE:** Corrispondenza UNIVOCA fra rappresentato e rappresentante

- Vantaggi: rappresentazione che si presta anche ad elaborazioni complesse, in quanto discretamente robusta e si accorge degli errori;
- Svantaggi: rappresentazione poco fedele ed intuitiva;

3) **RAPPRESENTAZIONE BINARIA:**

- Vantaggi: ESTREMAMENTE ROBUSTA ed economica;
- Svantaggi: limitata, non da' molte informazioni;

CARATTERISTICHE COMPLEMENTO A DUE

Vantaggi:

- elimina l'ambiguità sulla codifica dello zero;
- consente di utilizzare la sola somma per effettuare tutte le operazioni di somma e sottrazione;
- consente di usare la sola somma anche per la codifica e la decodifica dei numeri;

Svantaggi:

- richiede di fare attenzione ai traboccamenti perché semplicemente continuando a sommare 1 a un numero positivo prima o poi ci si ritrova con un numero negativo !

ALGEBRA BOOLEANA

Nasce per operare su affermazioni che possono assumere solo due valori: VERO o FALSO.

Postulati:

1) Definita X come (grandezza) variabile dell'algebra booleana: (P1) $X = 0$ if $X \neq 1$ (P1') $X = 1$ if $X \neq 0$

2) Definito X il valore opposto di quello assunto dalla variabile X:

(P2) if $X = 0$ then $X = 1$ (P2') if $X = 1$ then $X = 0$.

Si è definita l'operazione di NEGAZIONE (NOT): X è il NEGATO di X

3) (P3) $0 * 0 = 0$ (P3') $1 + 1 = 1$

4) (P4) $1 * 1 = 1$ (P4') $0 + 0 = 0$

5) (P5) $1 0 = 0 1 = 0$ (P5') $0 + 1 = 1 + 0 = 1$

Con P3,P4 E P5 si sono definite le due operazioni di PRODOTTO LOGICO (AND) e di SOMMA LOGICA (OR).

PRINCIPIO DI DUALITA'

Ogni postulato P_i dell'algebra booleana ha una seconda versione P_i' che si ottiene sostituendo:

- ogni valore 0 con un valore 1 (e viceversa);
- ogni operatore con un operatore + (e viceversa).

TEOREMA DI DE MORGAN

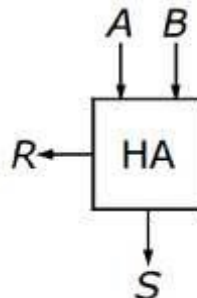
Consente di trasformare prodotti logici in somme e viceversa utilizzando i negatori.

SOMMA DI DUE NUMERI A 1 BIT (HALF ADDER)

A e B sono due numeri interi a 1 bit, S è il bit di somma, R è il bit di riporto.

A	B	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

TABELLA DELLE VERITA'

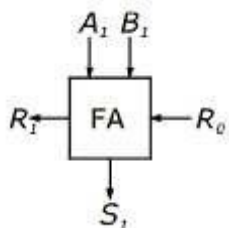


SOMMA DI DUE NUMERI A 2 BIT (FULL ADDER) A e

B sono due numeri interi a 2 bit:

- ciascuno dei numeri può valere 0, 1, 2 o 3;
- A0 e B0 sono i 2 bit meno significativi;
- A1 e B1 sono i 2 bit più significativi.

Servono a
Possono



BISTABILI

memorizzare un bit.
essere:

A ₁	B ₁	R ₀	S ₁	R ₁
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

1) TRASPARENTI: Quando il bistabile è abilitato, ogni variazione degli ingressi si ripercuote sulle uscite, dopo il tempo necessario alle porte logiche per commutare.

- Il bistabile si adegua prontamente alle variazioni degli ingressi.
- Le uscite risentono di ogni variazione degli ingressi, e ne propagano le conseguenze.

Il bistabile è trasparente: non filtra in alcun modo eventuali variazioni spurie degli ingressi.

Chiamiamo questi bistabili **LATCH**.

2) NON TRASPARENTI: Le uscite non risentono immediatamente di variazioni degli ingressi, ma solo al semiperiodo successivo del segnale di Clock.

Chiamiamo questi bistabili **FLIP-FLOP**.

CPU

Macchina di Von Neumann

La macchina di Von Neumann è costituita da:

- **CPU:** Unità master che gestisce in modo sequenziale il sistema.
- **MEMORIA DI LAVORO:** Contenitore di programmi da eseguire e dati su cui operare.
- **INTERFACCIA I/O:** Dispositivo elettronico che consente alla CPU di dialogare con le periferiche FUNZIONAMENTO DELLA CPU

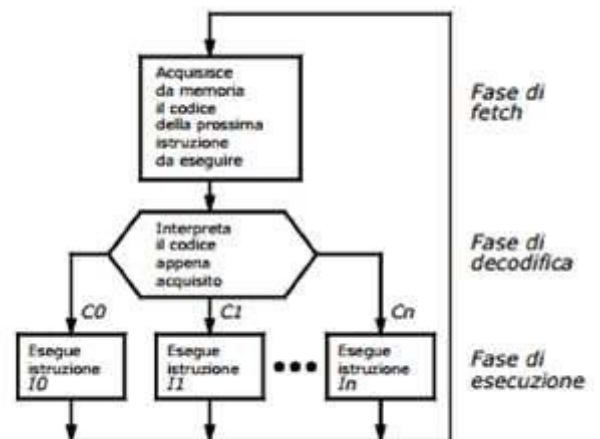
Fase di fetch:

- Preleva dall'esterno una stringa di bit che indica il prossimo passo da fare (macchina programmabile);
- Esegue, quindi, uno dopo l'altro una sequenza di passi (programma).

Fase di decodifica (decode): interpreta la stringa di bit come istruzione macchina.

Fase di esecuzione (execute):

- Svolge quanto richiesto;
- Accede all'esterno per scambiare dati.



La CPU segnala alla memoria:

- la cella a cui è interessata (mediante l'indirizzo);
- il tipo di operazione che intende svolgere (lettura o scrittura della cella).

CPU e memoria si scambiano il contenuto della cella, secondo la direzione richiesta dalla CPU.

Infatti la **CPU** ha il ruolo **Master** (decide quando e cosa fare), mentre la **Memoria** ruolo **Slave** (risponde alle richieste della CPU).

Interfaccia I/O

- Interagisce con la periferica secondo quanto richiesto dalla periferica stessa;
- Operazioni di lettura e scrittura nei registri diventano interazioni con il mondo esterno:

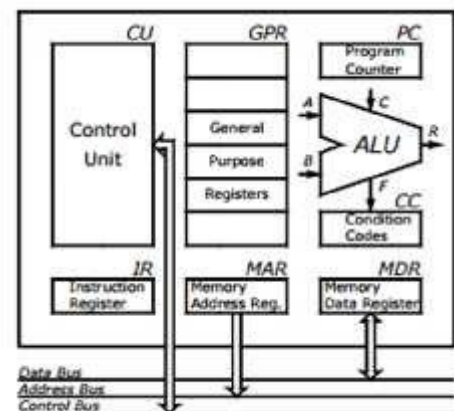
- invio di comandi alla periferica; •
- conoscenza dello stato della periferica;
- scambio di dati.

BUS

Ci sono 3 tipologie di bus:

- **DATA BUS:** Trasferisce "in parallelo" una stringa di bit fra Master (CPU) e Slave (Memoria o Interfaccia).
- **ADDRESS BUS:** Serve a indicare la cella cui la CPU intende fare riferimento.
- **CONTROL BUS:** È costituito da linee (fili) autonome, ciascuna con un proprio significato. Alcune linee (R, W) consentono alla CPU di gestire le interazioni con i dispositivi Slave. Altre linee consentono ai dispositivi Slave di poter attivare una interazione con il Master. Il numero di linee dipende dalla complessità della CPU.

LC2



Elementi costitutivi

- **PC (Program Counter):** registro che contiene l'indirizzo della cella di memoria nella quale si andrà a recuperare la prossima istruzione macchina da eseguire.
- **GPR (General Purpose Registers):** registri di uso generale, che contengono i dati (cioè le informazioni, codificate anch'esse mediante stringhe di bit) in corso di elaborazione.
- **CC (Condition Codes):** registro che contiene informazioni sull'esito dell'ultima elaborazione (es. risultato negativo, positivo ecc.)
- **IR (Instruction Register):** registro che contiene il codice (stringa di bit) dell'istruzione in corso di esecuzione.
- **MAR (Memory Address Register):** registro che consente alla CPU di emettere sull'Address Bus l'indirizzo della cella del dispositivo Slave che intende leggere o scrivere.
- **MDR (Memory Data Register):** registro che consente il trasferimento di un dato dalla CPU al Data Bus durante la scrittura nei dispositivi Slave, oppure dal Data Bus alla CPU durante la lettura dai dispositivi Slave.
- **ALU (Arithmetic Logic Unit):**

- effettua tutte le elaborazioni aritmetiche (somma in complemento a due, a volte sottrazione, moltiplicazione e divisione) e logiche (AND, OR, NOT, confronti);
- l'esito delle sue operazioni viene memorizzato nel registro CC (Condition Codes).
- **CU (Control Unit):**
- acquisisce e decodifica le istruzioni macchina presenti in memoria di lavoro;
- controlla il funzionamento di tutti gli elementi della CPU e (mediante il bus di controllo) del resto del calcolatore (dispositivi Slave).

ISA (INSTRUCTION SET ARCHITECTURE)

L'ISA è un insieme di attività elementari (istruzioni) che la specifica CPU è in grado di comprendere (decodificare) e svolgere. Ogni istruzione è caratterizzata da:

- codice operativo (**opcode**) che indica di quale istruzione si tratta;
- operandi (**operands**) che costituiscono i dati o le informazioni aggiuntive necessarie per eseguire l'istruzione.

Le principali operazioni sono:

- Operative: richiedono alla CPU di svolgere elaborazioni sui dati, utilizzando l'ALU (somme e sottrazioni, operazioni logiche, confronti, ecc.).
- Trasferimento: servono a prelevare da memoria di lavoro o da interfaccia di I/O i dati su cui operare e a trasferire in memoria di lavoro o interfaccia di I/O i risultati.
- Controllo: servono a variare l'esecuzione in sequenza delle istruzioni macchina (salti condizionati e incondizionati);

MODI D'INDIRIZZAMENTO

- **Immediato**: il dato è fornito nell'istruzione macchina.
- **Diretto**: l'istruzione macchina fornisce l'indirizzo della locazione di memoria contenente il dato.
- **Indiretto**: l'istruzione macchina fornisce l'indirizzo di una cella che contiene l'indirizzo della cella contenente il dato.
- **Base+offset**: l'istruzione macchina indica un registro GPR cui sommare un offset per ottenere l'indirizzo della cella.

LINGUAGGIO ASSEMBLY

Il linguaggio assembly è composto da:

LABEL OPCODE OPERANDS ;COMMENTS

Label: riferimento simbolico scelto dal programmatore per indicare l'indirizzo di memoria dell'istruzione.

Opcod: codice mnemonico dell'istruzione (ADD, JSR, ...).

Operands: riferimenti simbolici a registri o indirizzi di memoria. commento testo libero di spiegazione del significato dell'istruzione.

TIPI DI LINEE DI BUS E STADI USCITA

- Monosorgente: un solo dispositivo impone il valore alle linee, molti dispositivi lo acquisiscono (es. linee dell'Address Bus).

Esempio: Stadio di uscita **TOTEM POLE**

L'uscita può trovarsi in 2 stati: 0 a bassa impedenza, 1 a bassa impedenza. Il dispositivo pilota SEMPRE la linea cui è collegato.

- Multisorgente sincrona: a turno, diversi dispositivi impongono il valore alle linee (es. linee del Data Bus). E' possibile decidere a chi tocca (sincronizzare le linee).

Esempio: Stadio di uscita **TRISTATE**

L'uscita può trovarsi in 3 stati: • 0 a bassa impedenza; • 1 a bassa impedenza; • alta impedenza (Z). Serve un arbitro per decidere a chi tocca pilotare la linea di uscita.

- Multisorgente asincrona: diversi dispositivi impongono il valore alle linee. Non è possibile sincronizzare i dispositivi.

Esempio: Stadio di uscita **OPEN COLLECTOR**

L'uscita può trovarsi in 2 stati: • 0 a bassa impedenza; • alta impedenza (Z).

Il dispositivo pilota la linea cui è collegato solo se vuole imporre il valore 0.

RAM

Con la *Random Access Memory* non c'è rapporto causa-effetto tra un accesso a una cella e il successivo (ogni cella è ugualmente accessibile). Esistono due tipi di dispositivi:

– SRAM (Static RAM):

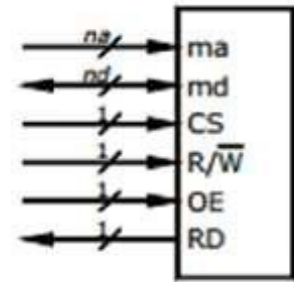
- Con stadi TRI STATE alle porte OR di colonna possiamo usare un Data Bus bidirezionale.
- Un segnale aggiuntivo (CS: Chip Select) consente di attivare o disattivare la risposta del dispositivo.

– DRAM (Dynamic RAM):

- I bistabili sono sostituiti da condensatori.
- Il valore del bit è associato alla carica presente sul condensatore.
- Se il bit vale 1, dopo un certo tempo il condensatore si scarica e "perde" l'informazione.
- È necessaria una attività periodica di refresh che ripristini la carica sui condensatori.

CHIP DI RAM

- na piedini di indirizzo monodirezionali (ma);
- nd piedini di dato bidirezionali (md);
- linea di input chip select (CS);
- linea di input read/write (R/W);
- eventuale linea di input output enable (OE);
- eventuale linea di output ready (RD).



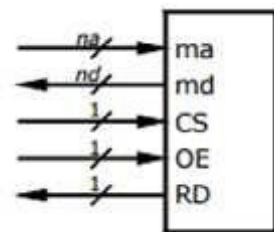
ROM

Memoria a sola lettura ROM (Read Only Memory) che mantiene il proprio contenuto anche in assenza di alimentazione:

- programma da eseguire all'accensione del calcolatore (fase di bootstrap);
- situazioni nelle quali il programma da eseguire è sempre lo stesso (applicazioni embedded).

CHIP ROM

- na piedini di indirizzo monodirezionali (ma);
- • nd piedini di dato monodirezionali (md);
- • linea di input chip select (CS);
- • eventuale linea di input output enable (OE); □ • eventuale linea di output ready (RD).



BANCO DI MEMORIA

Insieme di chip di memoria, che "riempie" una porzione dello spazio di indirizzamento della CPU considerata. Ogni "cella di memoria" del banco deve avere un numero di bit pari al numero di linee del Data Bus della CPU.

Il numero di celle di memoria del banco è tipicamente una potenza di 2. Il banco deve "apparire" come una sequenza di celle adiacenti in una determinata posizione dello spazio di indirizzamento della CPU.

SINCRONIZZAZIONE CPU – INTERFACCE I/O

Ogni operazione di I/O implica la sincronizzazione fra due riferimenti temporali:

- clock della CPU;
- "Orologio" dei fenomeni esterni, che agiscono sulla periferica collegata all'interfaccia. Tre modalità di sincronizzazione:
 - Direct Memory Access o DMA (indipendenti): L'interfaccia esegue autonomamente le operazioni di I/O, e avvisa la CPU solo a lavoro finito;

- Interrupt (vince l'orologio): La CPU esegue le operazioni di I/O quando l'interfaccia lo richiede, a seguito di quanto segnalato dalla periferica;
- Controllo di programma (vince il clock): La CPU esegue le operazioni di I/O quando il programma in esecuzione interagisce con l'interfaccia;

Caratteristiche Interrupt

Consente alla periferica di segnalare la necessità di servizio alla CPU: • serve linea dedicata del bus di controllo: INTREQ.

Particolarmente adatto a gestire fenomeni urgenti, che non possono attendere il tempo (casuale) di interrogazione di una soluzione a controllo di programma.

PIC (Programmable Interrupt Controller)

Circuito integrato di supporto alla gestione degli interrupt:

- L'interfaccia comunica al PIC la richiesta di interrupt.
- Se l'interfaccia è abilitata, il PIC attiva INTREQ.
- Quando riceve INTACK, il PIC comunica l'identificativo della periferica sul Data Bus.
- Se riceve più richieste di interrupt da diverse periferiche, dà precedenza a quella più prioritaria.

DATA PATH

Un registro può:

- emettere il suo contenuto sugli Output Internal Bus cui è collegato;
- campionare il valore presente sull'Input Internal Bus.

Se l'ALU è capace di propagare alla sua uscita O ciò che si presenta a uno degli ingressi A o B, le informazioni possono circolare fra i registri della CPU.

CONTROL UNIT (CU)

Input

- opcode dell'istruzione acquisita durante il fetch (L, S, A, B);
- situazione del registro CC (Z on/off);
- stato di avanzamento (n° dello step da eseguire).

Output

- comandi ai registri;
- comandi all'ALU;
- comandi al Control Bus.

Control Unit cablata (caratteristiche)

Vantaggi:

- Una struttura particolarmente efficiente.
- Assicura la massima velocità di esecuzione.
- Particolarmente adatta a CPU RISC: motivo del successo iniziale dell'approccio RISC.

Svantaggi

- Di difficile modifica.
- Di difficile utilizzabilità per macchine CISC, per l'esplosione della complessità del Logic Array.
- Solo recentemente, grazie all'evoluzione tecnologica, è applicabile a CU complesse.

Control Unit Microprogrammata (caratteristiche)

Vantaggi:

- Una struttura particolarmente regolare.
- Particolarmente adatta a CU complesse (CISC) con tecnologia hardware non evoluta.
- Sfrutta le tecniche di progettazione software.

Svantaggi:

- Problemi di velocità di esecuzione dovuti alla presenza della memoria di microprogramma.
- Soppiantata da un approccio cablato quando la tecnologia di integrazione lo consente.

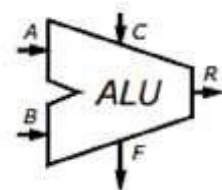
ALU

L'ALU (Unità aritmetico logica) è una componente che deve:

- provvedere a chiudere il Data Path fra i vari bus interni della CPU;
- svolgere operazioni logiche sugli operandi A e B:
 - AND bit a bit;
 - OR bit a bit;
 - NOT bit a bit.
- svolgere operazioni aritmetiche sui numeri A e B:
 - confronti;
 - somme (e sottrazioni) in complemento a due;
 - a volte moltiplicazioni e divisioni

Struttura

- A e B: ingressi operandi (parallelismo della CPU);
- R: risultato (parallelismo della CPU);
- C: comando (2C = numero di operazioni disponibili);
- F: flag di esito



Il “Carry look ahead”

Il riporto (i+1)esimo di una rete a 3 livelli che usa:

- una somma di prodotti (2 livelli);
- i cui termini sono somme o prodotti dei bit di dato (3° livello).

Si può dunque anticipare il riporto ("guardare avanti" il riporto) nei limiti della complessità di una rete combinatoria

MEMORIA CACHE (Memoria Associativa)

Serve solo a velocizzare gli accessi a memoria. La sua esistenza non è nota né al programmatore, né alla CPU (memoria “nascosta”).

Più veloce e leggera della DRAM.

POLITICA TAG ASSOCIATIVE

Considero la RAM divisa in blocchi e non in singole celle. I blocchi devono essere di dimensioni opportune.

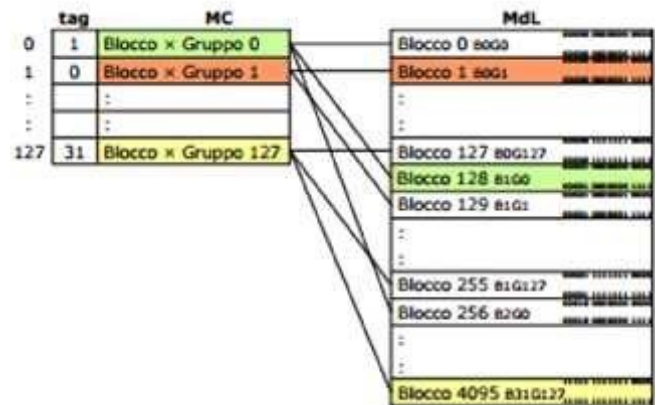
Vantaggi

Politica semplice:

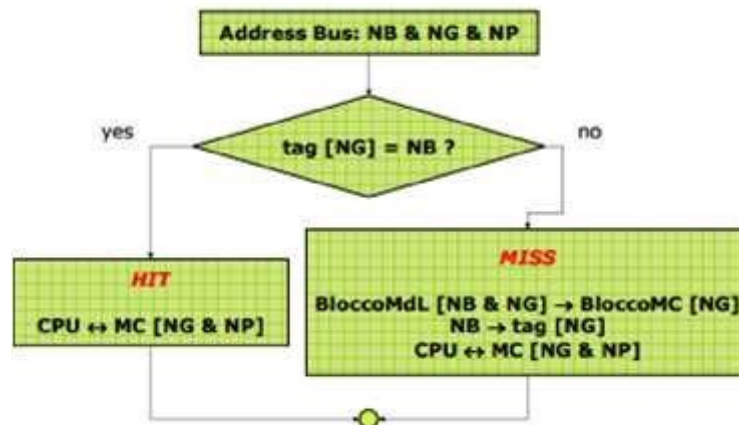
- il blocco richiesto dalla CPU può trovarsi solo in un blocco di cache: \Rightarrow la scoperta di HIT/MISS è rapida e priva di problemi;
- in caso di MISS, il blocco richiesto può essere ricopiato in un'unica posizione.

Svantaggi

- ogni blocco di MC ottimizza localmente l'accessibilità ai blocchi di MdL cui è assegnato;
- lo sfruttamento dei blocchi di MC non è uniforme.



Ricerca parola in cache Tag Associative



POLITICA FULL ASSOCIATIVE

Ogni blocco di MdL può andare a finire in qualsiasi blocco di MC.

Vantaggi

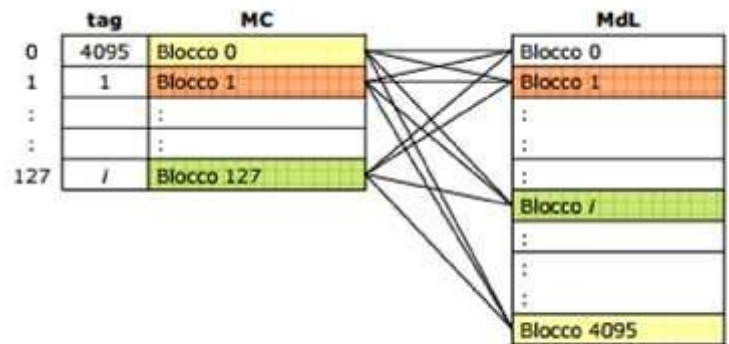
Politica ottimizzata:

- i blocchi presenti in MC sono sempre quelli che nel recente passato sono stati più richiesti dalla CPU;
- abbiamo un ottimo globale, con sfruttamento omogeneo dei blocchi di MC.

Svantaggi

Politica complessa e costosa:

- la ricerca del blocco richiesto implica il ricorso a memoria associativa per i tag;
- la ricerca del blocco di MC da sostituire implica l'uso dei contatori a saturazione, che devono anch'essi essere accessibili in modo associativo.



POLITICA SET ASSOCIATIVE

Politica “compromesso” tra le precedenti.

SCRITTURA IN MEMORIA CACHE (e relative politiche)

- **Politica store thru:** si modifica il dato sia in MC sia in MdL.
- **Politica store in:** si modifica il dato soltanto in MC.



CPU PIPELINE

Dividi la CPU in moduli più semplici o specializzati, quindi esegui le fasi quasi allo stesso tempo. La singola istruzione non è velocizzata ma il tasso di completamento è molto più alto.

Parallelismo temporale: prodotti diversi allo stesso istante. Avrò una Control Unit per ogni fase (fetch, decode, execute, rightback)

DIPENDENZE

1. **Di Controllo**: le istruzioni di salto “annullano” la pipeline.
 - **Speculative execution**: quando mi trovo ad un bivio e non so ancora dove andare, inizio a seguire la strada che ritengo più probabile.
 - **Branche predictor table**: è un elenco che tengo aggiornato con tutti i salti che ho fatto. La table deve essere costruita in modo che sia ad accesso associativo.
2. **Data Dependency**: uso una GPR, che viene modificata dall’istruzione precedente..Prevede l’uso delle bubble così da rallentare il flusso.
3. **Resource Dependency**: Doppie richieste di risorse contemporaneamente. Si crea una coda, tipo introduzione di una bolla, e il tempo raddoppia.

MACCHINA DI HARWARD

Separo MAR, MDR e memoria in VARIABILI E ISTRUZIONI.

Sono completamente indipendenti e possono essere usate contemporaneamente.

Svantaggi/Problema: Questa macchina va bene solo per i sistemi embedded.