Calcolatori Elettronici

Appunti delle Lezioni di Calcolatori Elettronici

Anno Accademico: 2023/24

Giacomo Sturm

Dipartimento di Ingegneria Civile, Informatica e delle Tecnologie Aeronautiche
Università degli Studi "Roma Tre"

Sorgente del file LaTeX disponibile al seguente link:
https://github.com/00Darxk/Calcolatori-Elettronici

Indice

1 Storia e Tipologia dei Calcolatori **1**
1.1 Evoluzione delle Architetture 1
1.2 Legge di Moore 2
1.3 Tipologie di Processori 2

2 Sistemi di Numerazione Binaria **6**
2.1 Rappresentazione a Virgola Fissa 6
2.2 Rappresentazione a Virgola Mobile 8

3 Organizzazione Generale di un Calcolatore **8**
3.1 Ciclo di Fetch-Decode-Execute 8
3.2 Parallelismo 10

1 Storia e Tipologia dei Calcolatori

1.1 Evoluzione delle Architetture

Un calcolatore è un oggetto che fornisce un risultato, dato un insieme dei dati inseriti.

L'evoluzione delle architetture dei controllori elettronici si è svolta principalmente negli ultimi settant'anni. Il processo complessivo che ha portato alla nascita dei calcolatori moderni viene divisa in generazioni. Si chiama generazione zero, l'insieme di calcolatori analogici, progettati per risolvere semplici operazioni, ideati da Pascal e Leibniz.

Il primo calcolatore programmabile venne ideato da Charles Babbage. Costruì prima una macchina differenziale in grado di calcolare funzioni polinomiali, mentre progettò la prima macchina programmabile, completamente analogica, in grado di leggere un input scritto su piastre di rame, e fornire un output, sempre su piastre di rame, utilizzando i dati e le operazioni inserite in input. Per poter operare su questi dati di input per fornire istruzioni alla macchina è necessario un linguaggio di programmazione, e la prima persona che ha tentato di implementare il linguaggio di Babbage fu Ada.

Il passaggio seguente in questa evoluzione venne trainato principalmente dai fondi bellici per realizzare macchine elettromeccaniche. La prima generazione si indica il periodo dove vennero realizzati calcolatori abbandonando componenti meccanici. La prima macchina del genere venne create da Alan Turing per decifrare il codice Enigma, realizzata tramite valvole, chiamata Colossus.

Dopo la guerra non servirono più a scopi bellici, per cui si tentò di vendere calcolatori sul mercato, creando la prima società di sviluppo e vendita di calcolatori sul mercato, Eniac.

Negli anni '50 John von Neumann descrisse l'idea di un calcolatore moderno, dove i dati vengono memorizzati su degli indirizzi di memoria.

L'IBM cominciò la sua storia vendendo calcolatori nel 1953, e continuo ad essere rilevante in questo ambito fino agli anni '80. In queste macchine ogni elemento viene definito dal termine "word", composto da un certo numero di bit.

La fine degli anni '50 e l'inizio degli anni '60 vide l'avvento dei transistor, utilizzati in questi anni per la creazione di calcolatori basati su transistor, la società rivale della IBM che venne creata in questo periodo fu la DEC. Utilizzando transistor vennero diminuiti i costi, e venne introdotta l'idea di utilizzare uno schermo grafico per interagire con l'utente. Il primo calcolatore costruito dalla DEC, PDP-1, fu il primo calcolatore di massa.

Per accedere ad un calcolatore si utilizzavano dei terminali, tramite un canale di comunicazione chiamato bus, per permettere anche la comunicazione tra elementi interni al calcolatore prodotti tra società differenti. Si parla comunque di grossi calcolatori per applicazioni scientifiche, militari o di pubblica amministrazione, chiamati "mainframe", dove ogni utente accedeva al calcolatore tramite terminali.

Fino agli anni '80 non vennero introdotti cambiamenti radicali all'architettura dei calcolatori, invece i miglioramenti di questi periodi ai calcolatori riguardarono soprattutto l'ottimizzazione del software e dell'hardware.

I primi "Personal Computer" vennero introdotti negli anni '80, dall'IBM, che fornì pubblicamente l'architettura del calcolatore. In seguito aumentò in enorme maniera l'utilizzo di PC, trainato

dall'aumento delle capacità della CPU, e dalla diminuzione dei costi delle memorie principali e secondarie.

La maggior parte dei dispositivi moderni contengono microcontrollori, piccoli processori, distribuiti in un contesto completamente pervasivo, su ogni dispositivo collegato ad una qualche fonte di energia. Questo concetto viene chiamato anche dell'"Internet of Things".

1.2 Legge di Moore

Uno dei fondatori dell'Intel, Moore, negli anni '60 definì empiricamente l'omonima legge, secondo cui il numero di transistor su un chip, CPU, memoria, etc., raddoppia ogni 18 mesi. Questo corrisponde ad un aumento del 60% all'anno.

L'evoluzione reale sembra aver seguito l'andamento descritto da Moore, ma recentemente l'evoluzione sta rallentando, a causa dei limiti fisici nella realizzazione dei transistor. Per cui esiste un limite superiore al numero di transistor su un unico chip. Per misurare la quantità di transistor su un singolo chip si utilizza la grandezza "Livello di Integrazione", si riescono a creare chip con un livello di integrazione nell'ordine di grandezza dei nanometri, ma livelli di integrazione superiore sono difficilmente realizzabili. Il limite teorico per memorizzare un bit di informazione corrisponde allo spin di un elettrone, per cui ci si aspetta una riduzione in questo andamento nei prossimi anni.

Oltre alla legge di Moore son presenti diverse statistiche per misurare l'evoluzione tecnologica dei processori. Dagli anni 2000 si utilizzano più di un core su un unico processo, introducendo semplice forme di parallelismo. Uno dei motivi principali per cui vennero introdotte queste architetture deriva dal limite alla frequenza di funzionamento di un processore, poiché all'aumentare della frequenza aumenta il calore prodotto da un processore. Le frequenze maggiori raggiunte da processori si trovano nell'ordine dei GHz, queste producono calore fino a 100 Watt. Per aumentare le prestazioni senza aumentare la frequenza, si introducono quindi forme di parallelismo nei processori. Una forma semplice consiste nella duplicazione dei componenti, oppure della "pipeline", che realizza le stesse prestazioni senza introdurre parallelismo fisico. Gia dal 2000 quindi la frequenza operazionale dei processori è rimasta costante, ed ha subito una leggera diminuzione, allo stesso modo del calore generato da un singolo processore. Anche se è possibile realizzare processori mono-core che lavorano a frequenze molto elevate, il costo associato al raffreddamento dei componenti non lo rende un approccio economicamente attuabile.

Dal punto di vista tecnologico per migliorare le prestazioni, bisogna cercare forme diverse di realizzazione di processori, che non utilizzano transistor, una di queste possibili tecnologie riguardano la computazione quantistica.

La legge di Nathan afferma che il software è un gas, riempie sempre completamente qualsiasi contenitore in cui viene inserito. Per cui molto velocemente e facilmente un calcolatore diventa obsoleto, questo alimenta un circolo vizioso che spinge l'evoluzione tecnologica, e rappresenta quindi la legge di Moore.

1.3 Tipologie di Processori

Un calcolatore è un dispositivo in grado di ricevere dei dati, di memorizzare in piccola parte i dati, di elaborare i dati, e di produrre un output. In generale un qualsiasi d dispositivo elettronico in gra-

do di soddisfare queste quattro specifiche può essere considerato un calcolatore. Si possono quindi definire diverse classi di calcolatori o processori sulla base delle loro prestazioni, all'aumentare delle prestazioni aumenta quindi il costo associato ad un dato processore. Esistono calcolatori monouso o "usa e getta", e microprocessori di basso costo, utilizzati negli elettrodomestici, automobili, o altri oggetti che non richiedono di capacità di computazione elevate, e soddisfano compiti specifici. Processori più evoluti, ma sempre specializzati, vengono utilizzai per applicazioni "mobile", oppure per piattaforme di gioco. L'unica differenza rispetto ad un Personal Computer è la loro specializzazione, mentre i processori di questa categoria svolgono applicazioni più generali "General Purpose", in grado di essere programmati. Processori ancora più avanzati vengono utilizzati per fornire servizi, non per l'elaborazione personale, e vengono chiamati server, ma in termini di tecnologia non presenta differenze evidenti rispetto ad un PC. Veniva utilizzati processori ancora più potenti, chiamati "Mainframe", sulla base della centralizzazione della computazione, dove un singolo processore soddisfa le richieste di tutti gli utenti, ma non vengono più utilizzati a favore dell'elaborazione distribuita.

Processori usa e getta come gli RFID "Radio Frequency IDentification" rappresentano la categoria di processori più diffusa, sono tipicamente passivi, senza batteria, ma esistono dispositivi attivi, di dimensione molto contenuta, nell'ordine di qualche millimetro, contenente un piccolo processore dotati di un transponder radio. Contiene una memoria di 128 bit complessivi. Il transponder è in grado di ricevere segnali su una certa frequenza, inviato da un lettore, questo segnale radio fornisce ulteriormente l'energia necessaria per alimentare il processore che invia il numero memorizzato in memoria. Gli RFID attivi dotati di una batteria non necessitano di essere molto vicini al lettore per operare, uno di questi dispositivi è il "Telepass".

Microprocessori sono oggetti di plastica che contengono un processore, una piccola memoria, e forniscono un collegamento con l'esterno da vari piedini metallici. Necessitano di un'alimentazione esterna, su uno di questi piedini. Questi processori non sono programmabili, e vengono usati in applicazioni di controllo.

Processori specializzati, non estendibili, ma di prestazioni molto superiori ai microprocessori sono i "Game Computer", che presentano effetti grafici speciali, per cui in generale presentano un processore grafico specializzato "Graphical Processing Unit" o GPU, ed un software di base limitato. Oltre alla memoria di base chiamata RAM, contengono la memoria di video, per gestire la visualizzazione a schermo chiamata VRAM. Generalmente questi processori CPU o GPU lavorano a non più di 4 GHz, per fornire informazioni sulle prestazioni di un processore si considera la banda di un processore, che rappresenta il numero di operazioni effettuabili in un dato intervallo di tempo. Si usa l'unità di misura FLOPS "FLOating points Per Second" supponendo il caso peggiore, quindi operazioni su numeri a virgola mobile. In generale questi dispositivi presentano una banda nell'ordine dei tera FLOPS. Questi sistemi sono chiusi, quindi non è possibile aumentare le prestazioni aggiungendo ulteriori chip al dispositivo. Appartengono alla stessa categoria le applicazioni Mobile, che presentano processori anche a otto core, con frequenze inferiori, poiché non presentano un sistema di raffreddamento attivo, e contengono una batteria e non un'alimentazione costante, per cui si utilizzano queste frequenze per diminuire il consumo energetico del processore. I processori utilizzati nell'ambito Mobile appartengono alla famiglia ARM, questa non è una casa produttrice come Intel o AMD, ma rappresentano una categoria di processori che vengono realizzati da diversi produttori, poiché è un'architettura aperta, di cui sono note le specifiche ed il linguaggio macchina.

Questo modello di mercato si basa interamente sulle licenze vendute dalla casa produttrice ARM, per cui si quando si parla di un processore di questa famiglia, si include anche la casa produttrice che ha prodotto il processore. Recentemente la Apple ha esteso l'uso di processori ARM anche su applicazioni di Personal Computer. Su questi dispositivi le funzionalità I/O vengono fornite tramite un 'interfaccia grafica basate su touch-screen.

Il Personal Computer si riferisce alla disciplina dell'elaborazione personale dei dati. Sono processori di specifiche non molto diverse dalle precedenti, ma sono programmabili. Tutti questi dispositivi sono connessi alla rete, per cui appartengono all'Internet of Things. La differenza tra un PC ed un server è la disciplina secondo cui l'elaborazione dei dati non è personale, ma fornisce un servizio. Tipicamente questi servizi vengono fornite tramite diversi server che lavorano in parallelo secondo la disciplina COW "Cluster Of Workstation", collegati tramite una rete ad alta velocità, che presentano una ridondanza nella replicazione dei dati, in caso uno di questi server abbia un malfunzionamento. La tendenza al parallelismo è quindi presente non solo a livello microscopico sui singolo processori, ma anche a livello macroscopico utilizzando più server. Permettono di continuare ad erogare il servizio in caso di un malfunzionamento, ed è molto raro che la maggior parte dei server nel cluster falliscono contemporaneamente. La realizzazione di questi server segue la disciplina della scalabilità orizzontale, ovvero vengono aggiunti nuovi server all'aumentare degli utenti, quando invece è presente un unico server si parla di scalabilità verticale, dove per fornire servizi a più utenti si aumentano le prestazioni di un unico server.

Esiste una tendenza di molte organizzazione a non realizzare un sistema di elaborazione con risorse proprie, ma utilizzare risorse nel Cloud, un'esempio molto diffuso è l'AWS, o gli Amazon Web Service, che forniscono memoria, memorizzazione di dati, e capacità di computazione accessibile nel Cloud. In questo approccio si sta ritornando all'approccio del Mainframe, ma in questo caso il terminale di accesso alle risorse fornite nel Cloud è anch'esso un calcolatore avente risorse di calcolo proprie, per elaborare una parte dei dati "In Premise".

Verranno trattati tre diversi tipi di processori, appartenenti alla famiglia Intel e ARM, ed un processore appartenente alla famiglia dei microcontrollori della famiglia AVR.

Il primo processore commercializzato dalla Intel è il 4004, con una frequenze tra le frazioni di un MHz, ed in grado di gestire poche centinaia di bit di memoria. Una variante di questo processore, specializzato per microcontrollori il 8008, fornì la base per la creazione del primo processore "general purpose" su un circuito integrato, il 8080.

La prima cifra nel nome corrisponde al tipo di architettura del processore, indica il numero di bit in cui vengono salvati e gestiti i dati sui registri del processore. A partire dagli anni '90 si cominciò ad utilizzare nomi diversi dai numeri per indicare i processori. Si introdussero memorie cache, ed all'inizio degli anni 2000 si introdussero diverse forme di parallelismo fisico, e non solo, mantenendo le frequenze inferiori ai 4 GHz. Per molti anni si utilizzava lo stesso processore anche per la gestione dello schermo, ma già da parecchi anni è la norma utilizzare due processori separati. Tutti i processori moderni della stessa famiglia sono compatibili con lo stesso linguaggio macchina. La denominazione di un processore indica le sue prestazioni, e sono quindi destinati a diversi settori di mercato, per cui l'evoluzione non dipende più dal nome del processore, ma dalla generazione dei processori. Attualmente ci si trova in una generazione intermedia tra la tredicesima e la quattordicesima, in generale un salto generazionale viene definito dal livello di integrazione, i processori moderni hanno un livello di integrazione di 7 nanometri. Sono tutte architetture che presentano

fino ad otto core, ma recentemente invece di utilizzare core identici sullo stesso processore sono state introdotte architetture ibride i cui core sono di almeno due forme diverse chiamati "p-core", per le prestazioni in termini di calcoli complessi, e gli "e-core", sono più efficienti in termini di consumo di energia. POssono avere fino a 24 stadi di pipeline, che permettono di avere altre forme di parallelismo, senza utilizzare parallelismi fisici.

Utilizzando un'unica catena di produzione, in base alla qualità in cui vengono prodotti si ottengono diversi processori, disattivando le componenti che non funzionano correttamente sul processore, e si vendono quindi come dei processori aventi prestazioni minori rispetto ad una versione completamente funzionante.

Il processore Intel Core i7 presenta sei core abilitati su otto core disponibili, la sua versione completamente abilitata corrisponde al processore Xeon. Presenta poco più di un miliardo di transistor ed un livello di integrazione nell'ordine dei 22 nanometri.

La società Acorn inventò negli anni '80 questo tipo di architettura bastata sui principi RISC (Acorn RISC Machine). Venne usato sui primi tablet prodotti dalla Apple. Nasce come un processore integrato ed a basso consumo energetico. Presenta un modello di commercializzazione diverso rispetto al resto del mercato, utilizzano un'architettura aperta, che permette a diverse case produttrici di realizzare questi processori, vendendo le licenze per poter produrre il processore. Ogni processore ARM viene quindi accompagnato dal nome dell'azienda che l'ha realizzato.

Si analizzerà in seguito l'Nvidia Tegra 2, un SOC "System Of a Chip" contenente due processori della famiglia ARM, una piccola GPU ed ulteriori componenti.

L'architettura AVR corrisponde a processori progettati per elettro-domestici, e per funzioni specifiche. Nacque da un progetto universitario del NIT nel 1996, dal nome dei suoi creatori Alf and Vergard RISC Processor. Presenta lo stesso pinout dell'8051 Intel. Presenta vari timer, un orologio interno, trasmettitore di impulso, interfaccia di sensori, convertitori analogico-digitali, transponder e comparatore di tensioni. Presenta memorie nell'ordine delle centinaia dei kilobyte per la memoria persistente Flash, una memoria programmabile da poche migliaia di byte "EEPROM", ed una memoria principale fino ad un massimo di 16 KB, per i microcontrollori più potenti.

2 Sistemi di Numerazione Binaria

Quando si misura la memoria si utilizza l'unità di misura byte, corrispondente a 8 bit, e si tende ad utilizzare potenze di due invece di potenze di dieci, utilizzando i bit quando si considera una velocità.

Si utilizza questa notazione per la struttura fisica della memoria, la cua dimensione viene definita dal numero di bit di un indirizzo binario. Dato un indirizzo definito da n bit, sono possibili 2^n indirizzi distinti di memoria. Per cui è più semplice lavorare con potenze di due quando si analizza la memoria.

Nei sistemi di numerazione binaria è presente una differenza sostanziale tra il concetto di numero, un entità astratta, ed il concetto di numerale, una sua possibile rappresentazione in un dato sistema di numerazione. Nell'ambito dei calcolatori elettronici il numero di caratteri diversi per poter rappresentare un numero è finito, poiché lo è la dimensione dei registri di memoria. Per cui i numeri possono essere rappresentati a precisione finita, e si perdono alcune proprietà come la chiusura rispetto alle sue operazioni, sono presenti errori di arrotondamento di un numero, ciò corrisponde ad un errore di "overflow". Inoltre non è possibile rappresentare tutti i numeri reali, essendo infiniti, per cui possono essere rappresentati solo numeri con un numero finito di cifre decimali in un dato intervallo.

2.1 Rappresentazione a Virgola Fissa

In generale un numero viene rappresentato rispetto ad una base b, dove ciascuna cifra a_i del numerale rappresenta il coefficiente di una potenza della base b:

$$N : a_m \cdots a_0, a_{-1} \cdots a_{-k}$$
$$N = \sum_{i=-k}^{m} a_i b^i$$

Questa rappresentazione viene chiamata posizionale. Per rappresentare un numero utilizzando una base b, sono necessari b simboli distinti per poter rappresentare tutte i possibili coefficienti. Dopo aver esaurito le dieci cifre arabe $0, \cdots, 9$ bisogna utilizzare diversi caratteri, per basi esadecimali si usano lettere dell'alfabeto per completare l'alfabeto di simboli utilizzato $A, \cdots, F$. Ogni numerale deve essere quindi associato alla sua base per poter ricavare le informazioni del numero che rappresenta.

Questa rappresentazione si dice a virgola fissa, poiché viene definito a priori tramite il parametro k il numero di cifre dopo la virgola utilizzate.

Per rappresentare l'insieme dei numeri naturali, viene utilizzata il sistema posizionale, in notazione binaria con n bit si possono esprimere tutti i numeri nell'intervallo $[0, 2^n - 1]$. Si sfruttano quindi tutte le 2^n posizioni disponibili, dove devono essere rappresentati per ogni numerale anche gli 0 non significativi.

Le operazioni aritmetiche di base come la somma si svolgono cifra a cifra, portando il resto sulla cifra successiva, in caso il risultato non possa essere rappresentato utilizzando n bit si ha un errore di overflow, o trabocco, nella propagazione del resto. Anche la moltiplicazione viene effettuata cifra

a cifra tra i due numerali, poiché sono presenti solo due simboli in base due, i prodotti parziali sono pari a zero, oppure al moltiplicando, la somma tra i prodotti parziali si svolge come descritto precedentemente. Inoltre è possibile si verifichi un errore di overflow sulle moltiplicazioni, molto più facilmente rispetto ad una somma.

Il sistema posizionale include la rappresentazione di numeri con una parte decimale, può essere rappresentato utilizzando n bit, fissata la posizione della virgola nel numerale. Si possono così rappresentare numeri a parte decimale positivi. In questo modo è possibile effettuare le operazioni di somma allo stesso modo dei numeri naturali. Nella moltiplicazione non si possono rappresentare ulteriori cifre decimali, per cui si perde precisione effettuando questa operazione tra due numerali a virgola fissa. Moltiplicare un numerale per 2^n corrisponde a spostare la virgola di n posizioni a sinistra, mentre per 2^{-n} corrisponde a spostare la virgola di n posizioni a sinistra.

Per rappresentare i numeri con segni è necessario modificare il sistema posizionale, sono possibili diverse variazioni per ottenere questo risultato. Una di queste si indica come rappresentazione per modulo e segno, dove il primo bit del numerale indica il segno del numero, se è 0 è positivo, mentre se è -1 è negativo, mentre si usano $n-1$ bit per il modulo. In questo modo si riduce l'intervallo di rappresentazione, potendo rappresentare numeri su un intervallo simmetrico $[-2^{n-1}+1, 2^{n-1}-1]$, ma sono possibili due rappresentazioni per lo zero ± 0

La rappresentazione complemento a 1 si aggiunge uno zero a sinistra, si utilizza la rappresentazione posizionale per numeri positivi, mentre per numeri negativi si rappresenta il modulo nella rappresentazione posizionale, ed in seguito si complementa il numerale bit a bit. L'intervallo di rappresentazione coincide per la rappresentazione per modulo e segno $[-2^{n-1}+1, 2^{n-1}-1]$, ed allo stesso modo è presente una doppia rappresentazione dello zero.

Per evitare la doppia rappresentazione dello zero si utilizza il sistema della complementazione a due, una piccola variante del complemento ad uno. Se il numero è positivo si utilizza la rappresentazione posizionale, mentre per un numero negativo si rappresenta il suo modulo con la rappresentazione posizionale, si complementa bit per bit e si somma ad uno. In questo modo l'intervallo di rappresentazione con n bit coincide a $[-2^{n-1}, 2^{n-1}-1]$, ed è possibile un'unica rappresentazione dello zero. Per cambiare di segno di un numerale in questa rappresentazione è sufficiente complementare a due il numerale. Nel sistema a complemento a due per effettuare l'operazione di somma è sufficiente svolgere una somma bit a bit, si verifica un overflow se la somma tra due numerali dello stesso segno cambia il segno. Per svolgere l'operazione di differenza si cambia di segno il secondo numerale e si svolge una somma. Le moltiplicazioni tra due numerali in questa rappresentazione si svolgono tra i valori assoluti, e se necessario si complementa il risultato.

Esiste un'altra rappresentazione chiamata ad eccesso. Dati n bit, si definisce un numero noto chiamato eccesso, generalmente 2^{n-1}. Può rappresentare numeri sullo stesso intervallo della rappresentazione a complemento a due. Per rappresentare un numero si somma all'eccesso, ottenendo un numero sicuramente positivo per cui si può rappresentare utilizzando semplicemente il sistema posizionale. In pratica i numerali si ottengono da quelli della rappresentazione a complemento a due, complementando il bit più significativo. In questa rappresentazione i numerali sono disposti sequenzialmente dal più piccolo al più grande, semplicemente sommando uno.

2.2 Rappresentazione a Virgola Mobile

Nella codifica ASCII si utilizza un bit in più, chiamato bit di parità, per controllare eventuali errori nella trasmissione. Ogni numerale esadecimale viene associato ad un carattere $(0 - 1F)$, oppure un'istruzione di testo $(20 - 7F)$. In totale sono presenti 128 caratteri in questa codifica. Venne introdotto il codice Unicode, codificato mediante 16 bit, in grado di codificare 65536 elementi. Ma questo numero non basta per codificare diversi alfabeti che si basano s su ideogrammi, per cui si utilizza la codifica UTF-8, a lunghezza variabile basato su Unicode.

3 Organizzazione Generale di un Calcolatore

I dispositivi elettronici sono in grado di eseguire direttamente solo un numero limitato di istruzioni semplici, il linguaggio macchina che fornisce al calcolatore la sequenza di istruzioni da eseguire è il linguaggio al più basso livello, per cui non è adatto ale persone. Per poter fornire istruzioni ad un calcolatore bisogna utilizzare un linguaggio di programmazione, ed un compilatore che lo traduce in linguaggio macchina, eseguibile dal calcolatore. Un calcolatore è formato da un modulo dei registri interni al processore, essenziali per il suo funzionamento, zona di memoria diversa dalla RAM, memoria volatile esterna, collegata tramite un Bus. Altri dispositivi di ingresso o uscita vengono collegati al Bus, il principale è la memoria secondaria, non volatile, sullo stesso canale di comunicazione. La memoria secondaria, convenzionalmente realizzato tramite dischi rigidi, è un dispositivo di ingresso-uscita, poiché è possibile leggere e scrivere dati sul disco.

3.1 Ciclo di Fetch-Decode-Execute

All'interno della CPU sono presenti un numero di registri limitati, l'ALU "Arithmetic Logic Unit", ed un registro di uscita. Ogni istruzione esegue lo stesso percorso all'interno della CPU. I dati vengono salvati sul registro interno, vengono trasferiti tramite dei Bus interni sui registri di entrata ed in seguito inviati alla ALU e il risultato viene salvato su un registro di uscita. Questo ciclo macchina si ripete allo stesso modo per ogni singola istruzione eseguita dalla CPU. La CPU ripete continuamente una stessa serie di istruzioni; carica le istruzioni dalla memoria al registro delle istruzioni IR, "Instruction Register" in seguito il Program Counter, o PC, viene incrementato, poiché contiene l'indirizzo di memoria dove è presente la seguente istruzione. In generale le istruzioni vengono salvate sequenzialmente sulla memoria, per cui è sufficiente incrementare il PC per poter passare all'istruzione immediatamente successiva. Non sempre si viene eseguita l'istruzione immediatamente successiva, poiché sono possibili istruzioni condizionali che possono generare dei salti, inviando un'indirizzo di memoria diverso dal successivo al PC. Dopo aver aggiornato il PC, viene decodificata l'istruzione letta all'indirizzo contenuto nel PC. Queste istruzioni possono non richiedere espressamente dei dati, ma in caso lo richieda nella fase successiva si cerca l'indirizzo di memoria dov'è contenuto il dato su cui bisogna applicare l'istruzione. L'operando viene calcolato su un registro. Dopo aver salvato i dati, questi attraversano l'ALU, eseguendo l'istruzione, e si ripete questo ciclo cercando l'esecuzione successiva in memoria.

Per eseguire un'istruzione la CPU può effettuare un'esecuzione diretta, dove l'hardware stesso è costruito per eseguire una singola istruzione macchina. Invece è possibile un'interpretazione dove

ogni istruzione macchina viene scomposta in istruzione più semplici eseguibili dal processore. Per distinguere l'istruzione macchina dalle istruzione eseguibili dal processore, più semplici ed elementari, vengono chiamate microistruzioni. Queste microistruzioni non sono note e sono incorporate all'interno del processore, mentre le istruzioni macchina sono note, e vengono convertite all'interno del processore in una sequenza di microistruzioni. Il vantaggio dell'esecuzione diretta è la capacità di eseguire direttamente l'istruzione sull'hardware, ma necessita di un hardware abbastanza complesso da poterle eseguire, ma ha un'esecuzione molto più efficiente, a scapito della semplicità dell'hardware. Nel caso dell'interpretazione, avendo disaccoppiato le istruzioni macchina dalle istruzioni eseguibili dal processore è possibile avere un repertorio di istruzioni molto più esteso, quindi l'hardware è relativamente semplice e più compatto, è quindi possibile una maggiore flessibilità nel progetto.

Poiché l'efficienza è il parametro fondamentale su cui si basa la creazione di progetti informatici, il primo approccio è preferibile, poiché è molto più efficiente rispetto al secondo. Per cui anche se un processore utilizza interpretazione, se non è più efficiente rispetto ad un processore ad esecuzione diretta, allora è preferibile quest'ultimo, nonostante la sua complessità. Questi due architetture vengono chiamate RISC, "Reduced Instruction Set Computer", la prima, mentre CISC, "Complex Instruction Set Computer", la seconda. Il numero di istruzioni eseguibili è una conseguenza dell'uso di uno dei due approcci, per cui questa classificazione non dipende dal numero di istruzioni eseguibili da un processore, ma dal modo in cui vengono eseguite. Nelle architetture RISC di esecuzione diretta sono possibile un numero ristretto di istruzioni, eseguite sui registri, dove sono presenti istruzioni apposite per poter effettuare accessi in memoria, ed ogni istruzione segue un ciclo di macchina. Nelle architetture CISC le istruzioni vengono interpretate tramite un microprogramma, aumentando il numero di istruzioni eseguibili, queste istruzioni vengono eseguite in memoria, inoltre ogni istruzione essendo composta da molte microistruzione comprende più di un singolo ciclo di macchina. Le prime architetture furono del tipo RISC, in seguito i progettisti utilizzarono l'architettura CISC, considerata migliore, ma considerando la migliore efficienza delle architetture RISC i progettisti di sistemi veloci riconsiderarono l'approccio dell'esecuzione diretta verso gli anni '80.

Per la retrocompatibilità dei processori Intel, alcune istruzioni dei loro processori sono istruzioni CISC. Le ultime architetture commercializzate dalla Intel quindi contengono molte istruzioni e sono di tipo ibrido per mantenere questa retrocompatibilità.

I calcolatori moderni vengono progettati utilizzando l'approccio RISC, e sono utilizzate varie tecniche per massimizzare la banda, la velocità con la quale le istruzioni vengono eseguite in un'unità di tempo, misurate tramite FLOPS, nell'ordine dei TFLOPS (Tera-FLOPS) per calcolatori moderni avanzati. Per aumentare questa banda si introducono meccanismi di parallelismo fisico, o architetture super-scalari oppure basata su pipeline. Per semplificare la decodifica delle istruzioni vengono realizzate tramite formati molto regolari. Inoltre vengono limitati i riferimenti alla memoria, accessibile solo tramite le istruzioni LOAD e STORE, tramite componenti hardware dedicate. Inoltre si tende ad ampliare il numero di registri interni del processore.

Questi principi anche se proprio della filosofia RISC, vengono rispettati in parte anche dalle architetture CISC.

3.2 Parallelismo

Per aumentare le prestazioni di un calcolatore è necessario introdurre forme di parallelismo, poiché non è più possibile aumentare la sua frequenza. A livello fisico è possibile aumentare le componenti fisiche, utilizzando più di un core, a tutti gli effetti processori indipendenti sullo stesso chip che lavorano in parallelo. Alternativamente è possibile introdurre forme di parallelismo senza duplicare le componenti fisiche dell'hardware, che opera a livello delle istruzioni. Questo meccanismo viene chiamato Pipelining, rappresenta l'idea della catena di montaggio, in questo modo ogni istruzione eseguita dal processore viene divisa in fasi, ed ognuna di questa fasi viene eseguita da una componente diverse del processore. Ognuna di queste esecuzioni vengono chiamati stadi, a livello hardware, che effettuano queste operazioni. In questo modo più istruzioni possono essere eseguite contemporaneamente, avendo un'istruzione diversa ad ogni stadio, essendo componenti indipendenti. Viene quindi completata un'istruzione ad ogni ciclo, ad ogni segnale ci clock, e si guadagna un fattore pari al numero di stadi nella pipeline, capaci di lavorare in parallelo. Si chiama pipeline, storicamente, poiché rappresenta un tubo dove gli elementi inseriti escono ion ordine sequenziale ed osservando l'interno di un tubo ad un dato istante sono presenti più di un elemento al suo interno, a diversa distanza dall'uscita. L'efficienza della pipeline si misura tramite due parametri, la latenza e la sua ampiezza di banda. Questo metodo trova un compromesso tra queste due. La latenza è il tempo impiegato per eseguire ogni istruzione, poiché ognuno degli n stadi viene eseguito in un tempo di clock T, la latenza è del tipo $n \cdot T$. Mentre la banda viene notevolmente aumentata utilizzando la pipeline, poiché a regime ogni periodo del segnale di clock viene completata un'istruzione: T^{-1}. Avendo tempi di clock nell'ordine dei nano secondi si ottiene una banda di $1000 \cdot T^{-1}$ MIPS, Milioni di Istruzioni Per Secondo. Poiché anche se sono presenti molti stadi, a regime, viene sempre completata un'istruzione ogni periodo di clock. QUesto rappresenta una situazione ideale che non corrisponde alla realtà. Poiché sono presenti delle dipendenze tra un'istruzione ed un'altra, per cui per eseguire un'istruzione bisogna aspettare il termine dell'esecuzione di un'altra, creando dei "buchi" nella pipeline, diminuendo la banda attuale.

Dentro uno stesso processore è possibile introdurre un parallelismo ulteriore su un'unico core tramite architetture superscalari, che raddoppiano il numero di pipeline interne al singolo core, utilizzando una componente hardware dedicata che avvia più di un'istruzione insieme. Inoltre è possibile introdurre un parallelismo a livello degli stadi più complessi, per cui solo lo stadio più lento viene parallelizzato per aumentare la sua velocità.