

# ADE - Architetettura degli elaboratori

Elia Ronchetti

Marzo 2022

# Indice

<b>1</b>	<b>Introduzione e Argomenti</b>	<b>4</b>
1.1	Rappresentazione dell'informazione . . . . .	4
1.2	Circuiti logici . . . . .	4
1.3	Instruction Set Architecture (ISA) . . . . .	4
1.4	Linguaggio Assembly . . . . .	5
1.5	Datapath . . . . .	5
1.6	Gestione delle eccezioni . . . . .	5
1.7	Tecniche di gestione dell'ingresso/uscita . . . . .	5
1.8	Gerarchie di memoria: cache . . . . .	5
<b>2</b>	<b>Sistemi numerici</b>	<b>6</b>
2.1	Conversione tra basi . . . . .	6
2.2	Operazioni aritmetiche e Overflow . . . . .	6
2.3	Operazioni con segno . . . . .	7
2.3.1	Modulo e segno . . . . .	7
2.3.2	Operazioni aritmetiche con MS . . . . .	7
2.3.3	Complemento a 1 (CA1) . . . . .	7
2.3.4	Complemento a 2 (CA2) . . . . .	8
2.3.5	Operazioni aritmetiche con CA2 . . . . .	8
2.4	Operazione di Shift . . . . .	8
2.5	Rappresentazione Eccesso $2^n$ . . . . .	9
2.6	Rappresentazione con la virgola . . . . .	9
2.6.1	( . . . . .	9
2.6.2	Virgola mobile . . . . .	9
2.6.3	Errore assoluto ed errore relativo . . . . .	10
<b>3</b>	<b>Circuiti logici</b>	<b>11</b>
3.1	Porte Logiche . . . . .	11
3.1.1	AND . . . . .	11
3.1.2	OR . . . . .	11
3.1.3	NOT . . . . .	11

3.1.4	Port con più di due ingressi . . . . .	12
3.2	Porte logiche derivate . . . . .	12
3.3	Decoder . . . . .	12

# Capitolo 1

## Introduzione e Argomenti

### 1.1 Rappresentazione dell'informazione

- Sistemi numerici
- Rappresentazione dei numeri interi con e senza segno
- Rappresentazione dei numeri in virgola fissa e mobile
- Rappresentazione dell'informatica non numerica

### 1.2 Circuiti logici

- Reti combinatorie
- Reti sequenziali e FSM (Finite State Machine)
- Rassegna di circuiti notevoli (decoder, multiplexer, register file, ALU, etc.)

### 1.3 Instruction Set Architecture (ISA)

- schema di von Neumann
- CPU, registri, ALU e memoria
- Ciclo fondamentale di esecuzione di una istruzione (fetch/decode/execute)
- Tipi e formati di istruzioni MIPS32

- Modalità di indirizzamento

## 1.4 Linguaggio Assembly

- Formato simbolico delle istruzioni
- Catena di programmazione (compilatore, assembler, linker, loader, debugger, etc.)
- Pseudo-istruzioni e direttive dell'assemblatore
- Scrittura di semplici programmi assembly
- Convenzioni programmatiche (memoria, nomi dei registri, etc.)

## 1.5 Datapath

- Percorsi dei dati per le diverse classi di istruzioni
- Controllo del percorso dei dati con FSM

## 1.6 Gestione delle eccezioni

- Tassonomia di eccezioni in terminologia MIPS32
- Modifiche alla FSM di controllo, registro Cause

## 1.7 Tecniche di gestione dell'ingresso/uscita

- Controllo di programma
- Interruzione di programma
- Accesso diretto alla memoria

## 1.8 Gerarchie di memoria: cache

- Cache a mappature diretta
- Cache fully associative
- Cache n-way set associative

# Capitolo 2

## Sistemi numerici

Con il termine bit definiamo l'unità di misura dell'informazione. Un bit può assumere solo il valore di 0 o 1.

Combinando tra loro più bit si ottengono strutture più complesse, per esempio:

- byte, 8 bit
- nybble, 4 bit
- word, 32 bit

Una rappresentazione è un modo per descrivere un'entità

Il sistema numerico decimale:

- usa 10 cifre
- è un sistema posizionale: ogni cifra assume un valore diverso a seconda della posizione che occupa

**Confronto tra Basi**    Inserire immagine confronto

### 2.1 Conversione tra basi

Svolti esercizi di conversione tra basi

### 2.2 Operazioni aritmetiche e Overflow

- Addizioni e sottrazioni
- Svolti esercizi con Overflow, bit di carry

## 2.3 Operazioni con segno

Ci sono diverse modalità per rappresentare il segno in base 2

### 2.3.1 Modulo e segno

La rappresentazione modulo e segno divide i bit di rappresentazione in 2, nel caso di 8 bit, 7 sono utilizzati per rappresentare il valore assoluto e il bit più significativo (MSB - Most significant bit), quello a sinistra, rappresenta il segno, 0 positivo, 1 negativo

$$1|0000100 \quad (2.1)$$

Questa rappresentazione è semplice e con n bit totali, si possono rappresentare i numeri interi nell'intervallo, ma ha alcuni problemi

- Esistono 2 rappresentazioni diverse per lo 0
- Un bit tra tutti i bit disponibili viene speso per il segno e questo è uno spreco, riduce inoltre la capacità di rappresentazione

### 2.3.2 Operazioni aritmetiche con MS

Possiamo avere overflow solo quando

### 2.3.3 Complemento a 1 (CA1)

È un'altra modalità di rappresentazione dei numeri interi con segno. Come indica il nome stesso, questo metodo si basa sull'operazione di complemento

**Complemento** è l'operazione che associa ad un bit (o ad ogni sequenza di bit) il suo opposto, cioè il valore ottenuto sostituendo tutti gli 1 con 0 e tutti gli 0 con 1

**Esecuzione** è semplice e diretta

1. Se il numero da codificare è positivo lo si converte in binario con il metodo tradizionale
2. Se il numero è negativo basta convertire in binario il suo modulo e quindi eseguire l'operazione di complemento sul numero appena convertito

**Problema** ancora doppia rappresentazione dello 0

### 2.3.4 Complemento a 2 (CA2)

Anche qui il MSB è 0 se  $x$  è positivo e  $\text{MSB} = 1$  se  $x$  è negativo

#### Esecuzione

1. Se il numero  $X$  è positivo esso rimane invariato
2. Se il numero  $X$  è negativo

Così elimino la doppia rappresentazione dello zero

### 3 Metodi per il calcolo di CA2

1. Bau
2. MiaoM
3. Regola pratica

Distinzione tra Operazione CA2 e Rappresentazione CA2

- La rappresentazione - come sono organizzati i bit
- Il calcolo - procedura di trasformazione

### 2.3.5 Operazioni aritmetiche con CA2

#### Somma a Sottrazione

1. Si esegue la somma su tutti i bit degli addendi, segno compreso
2. Un eventuale riporto (carry) oltre il bit di segno (MSB) viene scartato
3. Nel caso gli operandi siano di segno concorde occorre verificare la presenza o meno di overflow (il segno del risultato non è concorde con quello dei due addendi)

## 2.4 Operazione di Shift

Consiste nello spostare (shit) verso destra (right) o verso sinistra (left) la posizione delle cifre di un numero, espresso in una base qualsiasi, inserendo uno zero nelle posizioni lasciate libere.

- Left equivale a moltiplicare il numero per la base
- Right equivale a dividere il numero per la base



## 2.5 Rappresentazione Eccesso $2^n$

Balzata alla grande dalla prof, riprendere dalle slide

## 2.6 Rappresentazione con la virgola

### 2.6.1 (

Virgola fissa) É il metodo più semplice, scegliamo dove mettere la virgola e la fissiamo Il problema è che in base a dove posiziono la virgola ho diverse capacità di rappresentazione della parte intera o frazionaria

- Più a destra, scarsa rappresentazione intera, alta rappresentazione frazionaria
- Più a sinistra, scarsa rappresentazione frazionaria, alta rappresentazione intera

Questo porta rigidità

### 2.6.2 Virgola mobile

- Usa un bit per rappresentare il segno  $s$
- Usa altri bit per rappresentare la mantissa  $m$
- Usa altri bit per codificare l'esponente  $e$

Seguendo lo standard IEEE 754 la suddivisione è effettuata nella seguente modalità

#### 32 bit

- Segno - 1
- Esponente - 8
- Mantissa - 23

**64 bit**

- Segno - 1
- Esponente - 11
- Mantissa - boh

Anche qua la prof va a 200 all'ora e se ne sbatte il cazzo che la gente non sta capendo una sega

**2.6.3 Errore assoluto ed errore relativo**

Rappresentando un numero reale  $n$  in virgola mobile si commette un errore di approssimazione

# Capitolo 3

## Circuiti logici

Nell'elettronica digitale sia ingressi che le uscite possono assumere solo i valori di segnale alto (1 per convenzione) o basso (0)

Un circuito o rete combinatoria è quel circuito

Un circuito o rete sequenziale

### 3.1 Porte Logiche

Le porte logiche sono i componenti elettronici che permettono di svolgere le operazioni logiche primitive oltre che a quelle direttamente derivate. Esse realizzano le operazioni principali dell'algebra booleana. Sono circuiti elettronici che dati dei segnali 0 e 1 in input producono un segnale in output ottenuto effettuando una operazione booleana sugli ingressi. Le porte logiche hanno  $n$  input e generalmente 1 output

#### 3.1.1 AND

Questa porta logica svolge l'operazione logica di AND tra due bit, detta anche **prodotto logico**.

#### 3.1.2 OR

Svolge l'operazione logica di OR tra due bit, detta anche somma logica.

#### 3.1.3 NOT

Svolge l'operazione logica di NOT su un bit, detta anche **negazione logica**.

### 3.1.4 Port con più di due ingressi

Ad eccezione della porta NOT, le altre porte logiche possono esistere anche ad  $N$  ingressi, queste porte svolgono l'operazione logica associata su  $N$  bit invece che su 2.

Nei circuiti si possono realizzare porte a  $N$  ingressi collegando a cascata tra loro porte a 2 ingressi.

## 3.2 Porte logiche derivate

Porta NAND  $\rightarrow$  svolge l'operazione di NOT sul bit risultante dell'operazione di AND.

Porta NOR  $\rightarrow$  svolge l'operazione di NOT sul bit risultante dall'operazione di OR.

Porta XOR Opera come disgiunzione esclusiva tra due input, quando i 2 bit sono uguali produce in output un 1. Le porte NOR e NAND svolgono la funzione di inverter, sono definite universali

## 3.3 Decoder

Componente elettronico caratterizzato dall'avere  $n$  ingressi e  $2^n$  uscite Solo 1 valore è attivo per ogni combinazione di input, quindi l'ingresso seleziona una delle uscite, l'uscita selezionata ha valore 1 tutte le altre 0.

Nella foto il pallino vuoto corrisponde a 0, mentre senza pallino corrisponde a 1.