1 Lezione del 03-10-24

1.1 Assembler a 64 bit

Finora abbiamo studiato il linguaggio assembler a 32 bit (registri estesi EAX, EBX, ecc...). Vediamo adesso alcune caratteristiche dell'assembler a 64 bit.

Nei processori a 64 bit Intel-AMD x86 abbiamo 16 registri generali a 64 bit, con prefisso R, e che quindi si indicano come RAX, RBX, ecc... Di questi si può indirizzare la parte estesa dei 32 bit meno significativi (EAX), i 16 bit meno significativi (AX), e gli 8 bit meno significativi (AL). Per RAX, RBX, RCX e RDX si possono inoltre indirizzare gli 8 bit precedenti ad AL, BL, CL e DL usando AH, BH, CH e DH, ma questo è sconsigliato in quanto ci sono diverse limitazioni (non sono compatibili col prefisso REX).

Una lista completa dei registri genrali è la seguente, inclusi i nomi dei sottoregistri di dimensione minore:

64 bit	32 bit	16 bit	8 bit	8 bit (legacy)
RAX	EAX	AX	AL	AH
RBX	EBX	BX	BL	BH
RCX	ECX	CX	CL	CH
RDX	EDX	DX	DL	DH
RSP	ESP	SP	SPL	
RBP	EBP	BP	BPL	
RSI	ESI	SI	SIL	
RDI	EDI	DI	DIL	
R8	R8D	R8W	R8B	
R9	R9D	R9W	R9B	
R10	R10D	R10W	R10B	
R11	R11D	R11W	R11B	
R12	R12D	R12W	R12B	
R13	R13D	R13W	R13B	
R14	R14D	R14W	R14B	
R15	R15D	R15W	R15B	

Ricordiamo poi i registri RIP, l'instruction pointer, e RFLAGS che è il registro dei flag.

1.1.1 Spazio indirizzabile

Tecnicamente con architettura a 64 bit si potrebbero indirizzare 2^{64} byte distinti, ma i processori moderni permettono di indirizzarne solo $2^{48} = 256$ TiB, con alcuni modelli più recenti che arrivano a $2^{57} = 128$ PiB. I 47 (o 56) bit occupati sono i meno significativi, e i restanti 16 (o 7) devono avere il valore del bit più significativo utilizzato. Questo significa che sono indirizzabili effettivamente due porzioni contigue ma separate fra di loro di memoria:

	48 bit	57 bit
Regione alta	`0000 0000 0000 0000`	`0000 0000 0000 0000`
	`0000 7fff ffff ffff`	`01ff ffff ffff ffff`
Regione bassa	`ffff 8000 0000 0000`	`fe00 0000 0000 0000`
	`ffff ffff ffff ffff`	`ffff ffff ffff ffff`

Lo spazio I/O, infine, è di $2^{16} = 64$ KiB locazioni.

1.1.2 Istruzioni

Le operazioni possono possono usare 1, 2, 4 o 8 byte per un operando (rispettivamente Byte, Word, Long e Quad).

Notiamo che non possiamo usare displacement o operandi immediati a 64 bit: siamo limitati a 32 bit. Per ovviare a questo problema esiste una versione alternativa della `MOV`:

1.1.3 MOVABS

- **Formato:** `MOVABS $const, destination`
- **Azione:** porta una costante a 64 bit (che ci permette di scrivere) in un indirizzo generale.
- **Flag:** nessuno.

Operandi	Esempi
Immediato	`MOVABS $0xffff8105402300ef, %RBX`
Memoria	`CALL 0x00ef0b2a, %RAX`
Registro	`CALL %RAX, 0x00ef0b2a`

In generale, in assember a 64 bit si usano registri con valori base di 64 bit, e poi si indirizza con displacement a 32 bit, che in complemento a 2 concedono $\pm 2^{32}$, ergo $\pm 2\text{GB}$ di memoria indirizzabile rispetto alla base.

1.2 Reti logiche

Una rete logica è un modello astratto di un sistema fisico, costituito da dispositivi tra loro interconnessi. Le informazioni vengono codificate da questi dispositivi attraverso fenomeni fisici che si presentano in due aspetti distinti (corrente forte / corrente debole, tensione forte / tensione debole, magnetizzazione / non magnetizzazione, ecc...).

1.2.1 Caratterizzazione di rete logica

Una rete logica è caratterizzata da:

- Un'insieme di N variabili di ingresso. Il loro valore all'istante temporale t si chiama stato di ingresso. L'insieme di tutti i 2^N stati di ingresso si indicherà come $X.X = \{x_{N-1}x_{N-2}...x_1x_0\}$.
- Un'insieme di M variabili di uscita. Il loro valore all'istante temporale t si chiama stato di uscita. L'insieme di tutti i 2^M stati di uscita si indicherà come $Z.Z = \{x_{M-1}x_{M-2}...x_1x_0\}$.
- Una legge di evoluzione che determina come le uscite si evolvono in funzione degli ingressi.

Possiamo classificare le reti logiche in base a 2 criteri riguardanti l'evoluzione nel tempo:

- **Presenza/assenza di memoria**:

  - **Reti combinatorie:** analoghe a funzioni matematiche, le loro uscite dipendono solo dai loro ingressi in un qualsiasi istanti t;
  - **Reti sequenziali:** lo stato di uscita dipende dalla storia degli ingressi precedenti, ergo sono reti con memoria.

- **Temporizzazione della legge di evoluzione:**
  - **Reti asincrone:** l'aggiornamento delle uscite avviene costantemente nel tempo;
  - **Reti sincronizzate:** l'aggiornamento delle uscite avviene ad istanti di sincronizzazione discreti nel tempo.

I modelli sono ortogonali, ergo possiamo avere qualsiasi delle 4 combinazioni di queste caratteristiche:

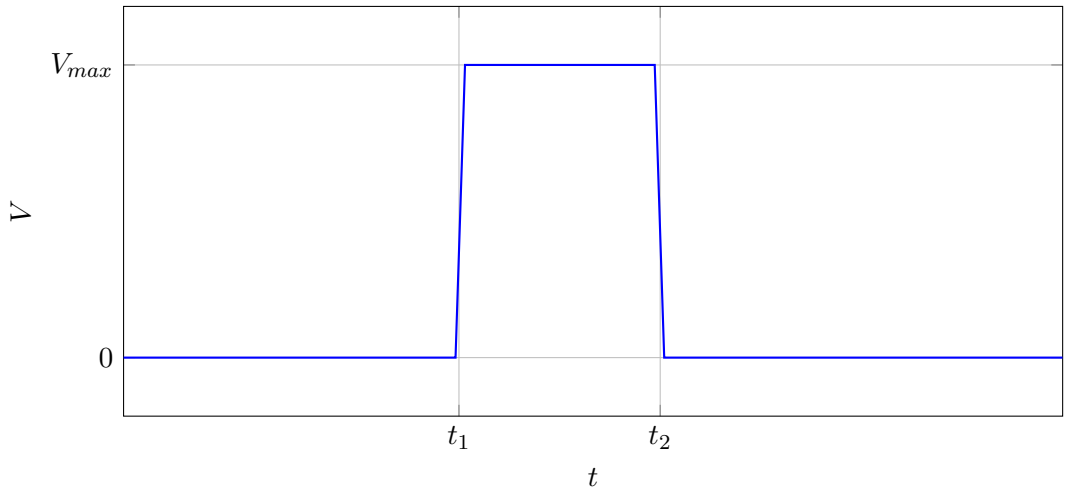
- Reti combinatorie (si considerano le sincronizzate come caso particolare);
- Reti sequenziali asincrone;
- Reti sequenziali sincronizzate.

Quindi in sostanza una rete logica comunica con l'esterno attraverso variabili logiche (0 e 1). L'interpretazione di questi messaggi è una convenzione del progettista, programatore, ecc...

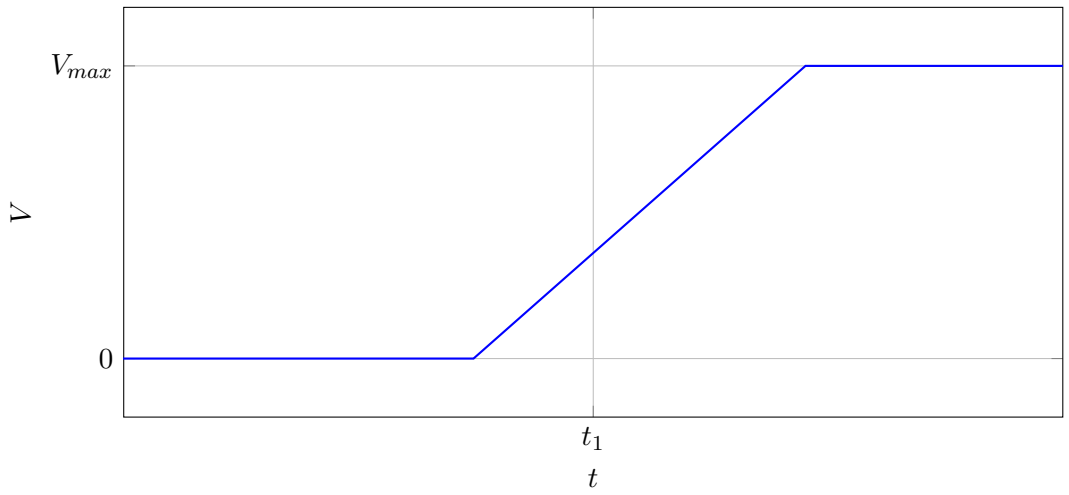
Usiamo le reti logiche per modellizzare circuiti elettronici all'interno del calcolatore, che codificano le informazioni in tensione. Notiamo quindi che una rete logica fisica ha, oltre agli ingressi e alle uscite, i collegamenti ai terminali positivi e negativi di un generatore di tensione, che noi ignoreremo.

1.3 Transizione dei segnali

Una variabile logica (per noi il voltaggio su un circuito) può settarsi (andare a 1), restare settato per tempi paragonabili a ΔT, e resettarsi (andare a 0) in un qualsiasi momento temporale t:

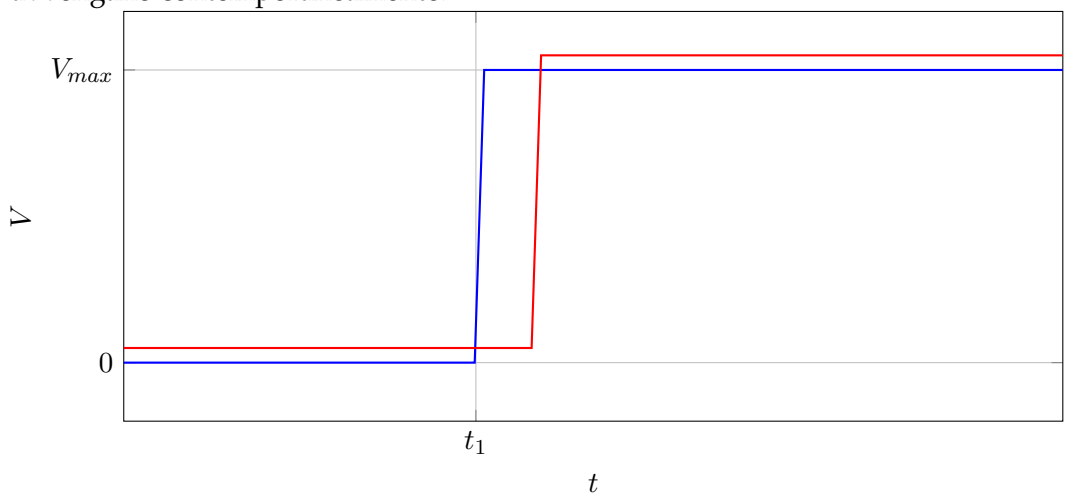


In un sistema fisico reale, durante la transizione c'è un periodo di indecisione in cui il voltaggio sale o scende fisicamente fino al valore necessario, sotto l'atto di una qualche potenza. Vediamo il grafico a $\Delta t << \Delta T$:



Decidiamo di ignorare questo problema, in quanto abbiamo visto che il Δt di transizione è molto più piccolo del Δt di stasi delle variabili.

Il problema si presenta nel caso si parli di **contemporaneità**. Supponiamo di avere una rete logica con due ingressi x_0 e x_1 e un'uscita z_0. Abbiamo che prima dell'istante t_1 lo stato di ingresso è $(1, 0)$, e che subito dopo lo stesso stato è $(0, 1)$. Nell'istante di transizione non abbiamo la sicurezza che le singole transizioni delle due variabili della rete avvengano contemporaneamente:



Questa considerazione è importante nel caso delle reti logiche asincrone, dove considerare le transizioni come contemporanee potrebbe portare alla comparsa di stati di uscita spuri, e nelle reti sequenziali, dove potrebbe portare ad evoluzioni imprevedibili del sistema.

1.4 Reti combinatorie

Una rete combinatoria è caratterizzata da:

- Un'insieme di N variabili logiche di ingresso;
- Un'insieme di M variabili logiche di uscita;
- Una descrizione funzionale $F : X \to Z$ che mappa stati di ingresso a stati di uscita;

- Una legge di evoluzione nel tempo che adegua $F(X)$ allo stato di ingresso X continuamente.

1.4.1 Tempo di attraversamento

Il tempo di attraversamento (o di accesso) è una caratteristica di tutte le reti logiche asincrone: è il tempo necessario perché la rete si "accorga" della variazione degli ingressi e aggiorni di conseguenza le sue uscite.

Questo tempo è solitamente non nullo, ed è quindi necessario attendere che la rete arrivi a **regime** prima di valutare le uscite. Questo vincolo prende il nome di **pilotaggio in modo fondamentale**: si dice che è una rete è pilotata in modo fondamentale quando chi la pilota aspetta sempre che essa arrivi a regime prima di valutare le sue uscite.