

PROGRAMMA

1 Introduzione

Sistemi a microprocessore – Livello hardware/software del corso. Segnali analogici e digitali. Computer e programma. CPU, Memoria e I/O. Storia del computer. Architetture di sistema: Von Neumann e Harvard. Bus. Sistemi general-purpose vs sistemi dedicati. Indirizzamento, potenze di due e di dieci (K, M, G, T). Bit, nibble, byte, word.

Rappresentazione dell'informazione – Numeri interi e codifica posizionale. Interi positivi: rappresentazioni in base 16, 10, 8 e 2. MSB e LSB. Cambio di base. Rappresentazione degli interi negativi: modulo e segno, complemento a 1, complemento a 2. Estensione del segno. Aritmetica binaria: addizione, moltiplicazione. Overflow. Caratteri alfanumerici: codifica ASCII e BCD. Conversione ASCII-base X e viceversa.

2 Reti Logiche

Teoria e tecnologia – Dal transistor al microprocessore: scale di integrazione. Stati logici e livelli elettrici. Tristate. Algebra di Boole. Rappresentazione di funzioni logiche: forma algebrica e forma tabellare. Porte logiche fondamentali: AND, OR, NOT. Enumerazione di funzioni ed altri operatori logici (XOR, identità, etc.). Forme canoniche: SP e PS. Proprietà dell'algebra, principio di dualità. leggi di De Morgan. Semplificazione di espressioni logiche. Completezza funzionale di NAND e NOR. Mappe di Karnaugh.

Reti combinatorie – Alcuni componenti combinatori: decoder binario, decoder BCD-7 segments, priority encoder, parity checker/generator, comparatore, multiplexer e suoi usi (selezione, sintesi di funzioni, cella logica flessibile, conversione parallelo-seriale, etc.), demultiplexer, tristate buffer e transceiver. Sintesi di Half- e Full-Adder, overflow detector, carry. Costruzione di una ALU. ROM: dimensioni, espansione di memoria, sintesi di funzioni, PROM e EPROM. Memoria (look-up table ROM) vs capacità di calcolo (ALU) nel progetto di una funzione aritmetico-logica.

Reti sequenziali – La funzione di memoria: automi a stati finiti. Diagramma di stato, rappresentazione tabellare e funzionale, schema a blocchi. Automi di Mealy e di Moore e loro equivalenza. Reti autonome. Ingressi e salti. Macchine a stati sincrone e a-sincrone. Il clock. Memoria e anelli: latch SR di NOR e di NAND. Flip-flop SR, JK, D, T: equazioni caratteristiche, condizioni di equivalenza, diagrammi temporali. Flip-flop edge-triggered e master-slave. Ingressi asincroni. Componenti sequenziali fondamentali: contatori e loro usi (misura di tempo, divisione di frequenza, conteggio UP-DOWN, etc.), registri, shift register e contatore ad anello. Esempi di sintesi sequenziale: sommatore seriale, moltiplicatore per numeri interi (parte operativa e parte di controllo). Dalle reti sequenziali dedicate al microprocessore.

3 Microprocessore

Architettura di un micro a singolo bus interno: IR, PC, MAR e MDR. Trasferimento dell'informazione tra registri. Esecuzione di una istruzione: ciclo fetch-decode-execute. Esempi di micro-sequenze di controllo: fetch di un'istruzione, load e store di un dato, operazione in ALU, salto incondizionato e condizionato. Architettura a tre bus interni. Esempio in pseudo-assembly: somma di N interi. Modi di indirizzamento: registro, diretto, immediato, indiretto, indicato, relativo, autoincremento, salto. Effective address. Lo stack e il suo uso: push, pop, chiamate a sottoprogrammi, passaggio di parametri. Prestazioni di CPU: tempo di esecuzione, MIPS, SPEC benchmarks. CPI. Miglioramento delle prestazioni: il ruolo della tecnologia e dell'architettura hardware/software. Pipeline ed esecuzione parallela. Il repertorio delle istruzioni. Macchine RISC e CISC: principali caratteristiche e differenze. Classificazione di processori: a stack, memoria-memoria, registro-registro, memoria-registro. Realizzazione di CPU: legame con la sintesi di reti sequenziali, sfruttamento della codifica incrementale degli stati, CPU cablate e microprogrammate, codifica orizzontale e verticale.

4 Memorie

Organizzazione gerarchica della memoria. MMU, memoria virtuale, protezione. Memorie RAM statiche e dinamiche: caratteristiche e differenze. Piedinatura e temporizzazione dei segnali in una DRAM; CS, OE, R/W, RAS e CAS. Colloquio col processore. Refresh. Cicli di attesa: come calcolarli, quale l'effetto sul tempo d'accesso. Costruzione di un banco di memoria DRAM. Memorie cache (a mappatura diretta): principio di località, struttura dell'indirizzo, calcolo del tempo d'accesso, rimpiazzamento di una linea. Byte-addressable computers: ordinamento little-endian e big-endian. Allineamento.

5 Input/Output

Tecniche e dispositivi – Interfacce di I/O e loro controllo software. Porte dati, di stato, di controllo. Decodifica degli indirizzi. Interfaccia parallela e seriale. Driver di I/O. I/O mappato in memoria e a spazio di indirizzamento separato. Modalità di I/O e loro caratteristiche: a controllo di programma, ad interrupt (polled e vectored), con DMA. Banda e latenza. Il disco rigido: funzionamento, tempo di accesso. DMA: registri interni, arbitraggio del bus, temporizzazione (Bus Request/Grant/Release), inizializzazione di un trasferimento DMA. Bus di sistema: bus host e chipset, bus locale (backplane), bus di espansione. Bus standards: PCI, SCSI, ISA. Calcolo dell'influenza dell'I/O sulle prestazioni del sistema: legge di Amdhal.

Interruzioni – Gestione dell'identità e della priorità. Interruzioni mascherabili e non, esterne ed interne, hardware e software (traps). Eccezioni. Differenze tra routine di servizio e subroutine. Vettorizzazione: tabella delle interruzioni, ciclo di accettazione di un'interruzione; IRQ, INTR e INTA, rivettorizzazione. Daisy chain e PIC. Altri usi delle interruzioni: nel debugger, nei sistemi operativi time-sharing.

6 Processore 8086/8088

Architettura – Un po' di storia. Piedinatura 8086/8088. Bus multiplexing. Cicli di bus. Modello di programmazione: registri general-purpose, puntatori, di segmento, flags, e loro uso tipico. Segment overriding. Lo stack con l'8086. Segmentazione della memoria: caratteristiche (dimensioni, sovrapposibilità), vantaggi e svantaggi. Indirizzo logico ed indirizzo fisico. Organizzazione della memoria: BHE, lettura ad indirizzi dispari. Struttura interna bipartita: EU e BIU. Coda di prefetch. Stato della coda e coprocessore matematico. Modo massimo e modo minimo. La fase di partenza (bootstrap). Il repertorio delle istruzioni. Modalità di indirizzamento: dati (nel codice, nei registri, in memoria, nell'I/O), salti (inter- e intra-segmento). Formati delle istruzioni: lunghezza variabile, tabelle di codifica, default registers, codifiche alternative. Evoluzione dell'8086: dal 286 al Pentium.

Linguaggio Assembly – La gerarchia del software: applicazioni utente, sistema operativo (DOS), firmware (BIOS). Compilatori e assembler: moduli sorgente, oggetto, eseguibile, di libreria. Assembler, linker, loader. Multiprogrammazione. Assemblatore a due passi: location counter, tabelle dei simboli, forward e backward reference. Rilocazione. I programmi 'asm', 'link', 'debug'. Disassemblaggio. Analisi di programmi assembly: file .lst, "dump" della memoria, debugging. Macro e subroutine: similarità e differenze. Istruzioni e direttive. Operatori. Parole riservate. Commenti. Definizione di segmenti: allineamento, combinabilità, classe. La direttiva ASSUME. Moduli e segmenti. Allocazione di memoria: variabili singole, stringhe, vettori. L'operatore \$. Simboli definiti dal programmatore: costanti, variabili. Definizione implicita ed esplicita di etichette. Cicli (LOOP). Definizione di procedure. Comunicazione tra moduli. Gruppi di istruzioni: aritmetiche, logiche, di trasferimento dati, di controllo del flusso, di I/O, etc. La tabella delle interruzioni: Intel, DOS e BIOS interrupts. Ritorno al DOS. Esempi di funzioni dell'int 21h del DOS: stampa a video di una stringa, input da tastiera, richiesta della data di sistema. Rivettorizzazione della tastiera (int 09h). Programmi TSR (int 27h).

Note

1. Testo di riferimento: Bucci, *Architetture dei calcolatori elettronici*, McGraw-Hill Libri Italia, 2001.
2. Testi consigliati: Hamacher, Vranesic, Zaky, *Introduzione all'architettura dei calcolatori*, McGraw-Hill Libri Italia, 1998; Patterson, Hennessy, *Struttura, organizzazione e progetto dei calcolatori*, Jackson Libri, 1999.
3. La pagina web del corso (www.dsi.unifi.it/~columbus/calco102.html) contiene puntatori a software assembler usato nelle esercitazioni, e ad altro materiale distribuito a lezione.