

1

Introduzione

L'evoluzione dei sistemi hardware digitali negli ultimi cinquant'anni è stata caratterizzata da miglioramenti in termini di funzionalità, costi e prestazioni mai visti in altri settori tecnologici. Oggi i sistemi hardware digitali pervadono le nostre vite, inseriti all'interno di oggetti di uso quotidiano. Questi sistemi, chiamati *sistemi dedicati*, sono tipologie di sistemi di elaborazione che svolgono compiti specifici all'interno di sistemi di tipo diverso. Tra gli esempi più comuni si possono ricordare i dispositivi mobili (palmari, cellulari), le macchine fotografiche o telecamere digitali, l'elettronica per applicazioni automobilistiche, i sistemi antifurto, elettrodomestici, apparecchiature per le telecomunicazioni, così come moltissime apparecchiature biomedicali.

La crescente diffusione dei sistemi dedicati all'interno di molti oggetti di uso quotidiano e la continua evoluzione delle tecnologie di miniaturizzazione rendono sempre più reale il concetto di *pervasive computing*, che richiede l'integrazione su singolo chip di sistemi complessi. Questi dispositivi includono componenti hardware digitali dedicati, processori (spesso più di uno) tra loro interconnessi, componenti software interagenti, sensori, attuatori per l'interazione con il mondo fisico circostante, componenti a radiofrequenza per la comunicazione, componenti meccaniche ecc.

Progettare questi sistemi è un compito difficile che richiede, oltre a delle solide competenze miste hardware e software, intuito ed esperienza. Tra le competenze di base richieste è quindi anche necessario essere in grado di realizzare componenti hardware dedicati, a partire dalla descrizione delle funzionalità richieste. Obiettivo di questo testo è quello di insegnare le tecniche fondamentali per progettare e implementare sistemi hardware digitali sincroni complessi.

Un sistema hardware è un sistema realizzato mediante interconnessione di componenti elettronici che possono essere analogici o digitali: gli ingressi e le uscite di un sistema digitale possono assumere solo valori finiti e discreti. Un sistema digitale sincrono è un sistema in cui le uscite degli elementi cambiano valore solo in istanti ben precisi di tempo. In un sistema asincrono invece, le uscite possono variare in modo arbitrario nel tempo, rendendone più complessa la gestione. Oggi, la stragrande maggioranza dei sistemi hardware digitali è sincrona.

È più semplice comprendere i sistemi hardware digitali considerando le modalità con cui vengono descritti, che possono essere distinte in base al livello di dettaglio della descrizione. In questo ambito consideriamo tre livelli: il livello *comportamentale*, il livello *logico* e il livello *circuitale*. Il livello comportamentale descrive in modo astratto gli ingressi, le uscite e la relazione tra di essi (il comportamento), ossia le funzionalità del sistema. Il livello logico considera la struttura del sistema definendo l'interconnessione (la rete) degli elementi logici di base, chiamati porte logiche, che rappresentano i componenti elementari utilizzati nella progettazione. A livello circuitale, i componenti di base sono rappresentati da dispositivi elettronici che implementano i componenti logici.

Questo libro è dedicato alla progettazione logica, ossia alla trasformazione di una descrizione comportamentale in una descrizione in termini di reti di porte logiche, senza occuparsi dell'implementazione delle porte logiche a livello circuitale. L'obiettivo è quello di insegnare le diverse modalità di rappresentazione a livello comportamentale di un sistema digitale e le tecniche che consentono di trasformare questi modelli in una descrizione a livello strutturale (la cosiddetta sintesi logica), ossia nella rete di porte logiche. In realtà, la relazione tra descrizione comportamentale e descrizione strutturale non è univoca, ma esistono più reti che realizzano lo stesso comportamento. Quindi, in genere, si vuole realizzare una rete logica che soddisfi anche dei vincoli non funzionali, quali prestazioni, dimensioni, costi, consumo di potenza, in base alle caratteristiche e all'applicazione finale del sistema. In particolare, questo libro considera come criteri per l'identificazione della soluzione migliore la dimensione della rete finale, ossia il numero di porte logiche, o le prestazioni, ossia il tempo massimo di attraversamento della rete dall'ingresso all'uscita.

Progettare un sistema significa quindi passare da una rappresentazione a un'altra, sempre più dettagliata, fino a ottenere il circuito fisico da realizzare. La rappresentazione iniziale consiste in genere in una descrizione in italiano, piuttosto imprecisa, dei requisiti funzionali e non del sistema da realizzare. Il primo passo consiste quindi nel tradurre questa descrizione in una rappresentazione più precisa e formale, iniziando dall'identificazione degli ingressi e delle uscite del sistema e della relazione che tra essi intercorre. La modalità di rappresentazione dipende dal tipo di funzionalità che si vuole descrivere: i sistemi più semplici sono caratterizzati dal fatto che le uscite dipendono dagli ingressi che si presentano nello stesso istante, mentre in quelli più complessi (la maggior parte dei sistemi) i valori delle uscite dipendono dalla sequenza temporale degli ingressi, ossia devono mantenere memoria del comportamento passato del sistema.

Le reti logiche che implementano una funzione in cui il valore delle uscite dipende dal valore corrente degli ingressi vengono chiamate reti combinatorie. Un

esempio è un addizionatore che realizza la somma tra due numeri presenti sugli ingressi. Le reti logiche che realizzano una funzione che dipende non solo dai valori correnti degli ingressi, ma anche dalla storia dei precedenti valori di ingresso, sono chiamate reti sequenziali. Tale dipendenza sarebbe molto difficile da soddisfare se il sistema dovesse effettivamente ricordare tutta la storia degli ingressi: in pratica i valori di ingresso corrente portano il sistema in un insieme finito di configurazioni, chiamate stati. Un sistema sequenziale prende quindi in ingresso la configurazione attuale dello stato e i valori di ingresso correnti e genera i nuovi valori di uscita e il nuovo stato. Dopo un certo intervallo di tempo il nuovo stato diventa lo stato corrente e il processo di calcolo del nuovo stato e dei nuovi ingressi si ripete. Nel nostro caso la configurazione del sistema cambia in risposta a un segnale di ingresso speciale, il clock, rendendo il sistema sincrono. Un esempio di semplice sistema sequenziale è quello di un semaforo, in cui il numero degli stati rappresenta la sequenza di cambiamento del colore del semaforo: rosso, giallo, verde, rosso.

Le modalità di rappresentazione delle funzionalità di un sistema combinatorio e di un sistema sequenziale sono diverse. Nel caso dei sistemi combinatori è possibile rappresentare la funzionalità mediante una tabella che elenca, per ogni possibile configurazione degli ingressi, il valore delle uscite. Queste tabelle sono chiamate tabelle della verità e sono una buona rappresentazione di funzioni con pochi ingressi. Per sistemi con un numero di ingressi più elevato, la tabella della verità cresce di dimensioni in modo esponenziale: una rappresentazione alternativa è la descrizione della funzione mediante una espressione composta da operazioni sugli ingressi. Nel caso dei sistemi sequenziali è necessario specificare, per ogni configurazione di ingresso e di stato corrente, il valore delle uscite e dello stato prossimo: si chiama in questo caso tabella di transizione dello stato.

In un sistema digitale, gli ingressi e le uscite sono segnali elettrici, che vengono rappresentati da valori discreti di tensione. In particolare, i sistemi digitali più semplici considerano solo due valori di tensione, rappresentati convenzionalmente da 0 e 1, ottenendo quindi un sistema binario: ogni ingresso o uscita può assumere solo due valori. I sistemi digitali binari rappresentano la base di quasi tutti i sistemi hardware digitali. Il principale vantaggio dei sistemi digitali è la rigorosa base matematica su cui si fondano, che deriva dalla logica matematica e dall'algebra di Boole. Questo è il motivo per cui il libro si apre con le basi dell'algebra di Boole, introducendo le operazioni logiche fondamentali che consentono di manipolare i valori logici. Il Capitolo 2 mostra che esiste una equivalenza tra tabelle della verità ed espressioni booleane, per cui è possibile derivare una espressione booleana da una tabella della verità e viceversa.

Come già accennato, gli elementi logici di base nella rete logica sono le porte logiche: il capitolo mostra anche che esiste una equivalenza tra espressioni logiche e descrizione in termini di rete di porte logiche, grazie al fatto che per ogni operatore logico esiste una corrispondente porta logica.

Il Capitolo 3 approfondisce le basi dei sistemi di codifica, in particolare i codici numerici, introducendo poi le operazioni aritmetiche, focalizzate sulla codifica binaria.

Dopo aver fornito le basi matematiche per la rappresentazione sia del comportamento sia della struttura dei sistemi digitali, il libro prosegue nella presentazione dei metodi di trasformazione di una rappresentazione comportamentale nella rete logica ottimizzata secondo i criteri di prestazioni e area. Il Capitolo 4 inizia considerando le tecniche per trasformare la rappresentazione comportamentale di un sistema combinatorio nella corrispondente rete logica minima, considerando come criterio la minimizzazione del numero di porte logiche. Le prime reti logiche considerate sono le reti a due livelli, ossia costruite a partire dalla rappresentazione canonica derivata della funzione booleana, che esprime la funzione come la somma logica di termini di prodotti logici delle variabili di ingresso (o a partire dalla forma canonica duale, chiamata prodotto di somme). Il processo di riduzione di una funzione booleana nella sua forma minima è chiamato minimizzazione booleana. Il libro presenta sia dei metodi semplici, applicabili a funzioni con un numero di ingressi molto piccolo, sia un algoritmo dettagliato che consente di ottenere in modo esatto l'implementazione minima, con il minor numero di porte logiche. Questo metodo, chiamato metodo di Quine McCluskey, al crescere del numero degli ingressi può rapidamente non portare a una soluzione. Il capitolo presenta quindi i metodi euristici utilizzati negli strumenti software di sintesi logica combinatoria, il cui obiettivo è quello di trovare una buona soluzione velocemente senza garantire la soluzione minima.

Infine il capitolo presenta le principali strategie per progettare reti logiche a più di due livelli, che consentono di minimizzare ulteriormente il numero di porte logiche rispetto alle reti a due livelli. Si tratta di strategie sofisticate, implementate in strumenti software di supporto alla progettazione, sia commerciali sia accademici, come SIS (sviluppato dalla University of California, Berkeley). Il Capitolo 5 conclude la parte relativa alle reti combinatorie, presentando come realizzare componenti combinatori più complessi a partire dalle porte logiche.

Il Capitolo 6 è il primo dei capitoli dedicati ai sistemi sequenziali, il cui comportamento dipende non solo dai valori di ingresso correnti ma anche dalle sequenze passate e che devono quindi memorizzare tale informazione, rappresentata dallo stato. Il capitolo introduce i modelli di rappresentazione a livello comportamentale, basati sulle macchine a stati finiti, e a livello strutturale. Per mantenere le

informazioni di stato è necessario disporre di elementi di memoria, realizzati a partire da un piccolo numero di porte logiche connesse in modo opportuno. Il Capitolo 7 presenta e classifica in base alle loro caratteristiche le diverse strutture degli elementi di memoria, che possono essere distinte principalmente in latch e flip-flop. Le porte logiche e gli elementi di memoria rappresentano quindi i componenti base per realizzare le reti logiche sequenziali.

Obiettivo del Capitolo 8 è quello di presentare i metodi per derivare la macchina a stati finiti a partire da una descrizione a parole e quindi trasformare tale descrizione in una rete logica sequenziale, introducendo i passi necessari alla sintesi sequenziale.

La fase di ottimizzazione della macchina a stati finiti è oggetto dei due capitoli seguenti. Il Capitolo 9 dedicato all'ottimizzazione della rappresentazione comportamentale mentre Capitolo 10 affronta l'analisi delle possibili ottimizzazioni della rete logica derivata dalla sintesi. Infine, il Capitolo 11 presenta le principali componenti sequenziali, utilizzati nella realizzazione di sistemi più complessi.

Al termine di ciascun capitolo (a eccezione dei Capitoli 5 e 11, che come si è detto sono di natura più descrittiva) sono riportati numerosi esercizi risolti e un certo numero di esercizi proposti. Le soluzioni di questi ultimi sono disponibili nel booksite abbinato al libro, all'indirizzo:

<http://www.apogeeonline.com/libri/88-503-2229-1/scheda>