

Carlo Brandolese
William Fornaciari

Sistemi embedded

Sviluppo hardware e software
per sistemi dedicati



Copyright © 2007 Pearson Paravia Bruno Mondadori spa
Via Archimede, 23 - 20129 Milano
Tel.: 02 74823.1
E-mail: hpeitalia@pearson.com
Web: <http://hpe.pearsoned.it>

Le informazioni contenute in questo libro sono state verificate e documentate con la massima cura possibile. Nessuna responsabilità derivante dal loro utilizzo potrà venire imputata agli Autori, a Pearson Paravia Bruno Mondadori spa o a ogni persona e società coinvolta nella creazione, produzione e distribuzione di questo libro.

I diritti di riproduzione e di memorizzazione elettronica totale e parziale con qualsiasi mezzo, compresi i microfilm e le copie fotostatiche, sono riservati per tutti i paesi.

LA FOTOCOPIATURA DEI LIBRI È UN REATO Le fotocopie per uso personale del lettore possono essere effettuate nei limiti del 15% di ciascun volume dietro pagamento alla SIAE del compenso previsto dall'art. 68, commi 4 e 5, della legge 22 aprile 1941 n. 633. Le riproduzioni effettuate per finalità di carattere professionale, economico o commerciale o comunque per uso diverso da quello personale possono essere effettuate a seguito di specifica autorizzazione rilasciata da AIDRO, corso di Porta Romana n. 108, 20122 Milano, e-mail segreteria@aidro.org e sito web <http://www.aidro.org>.

Copy-editing: Donatella Pepe
Impaginazione: Carlo Brandolese
Grafica di copertina: Gianni Gilardoni
Stampa: Tip.Le.Co. – S. Bonico (PC)

Tutti i marchi citati nel testo sono di proprietà dei loro detentori.

978-88-7192-342-0

Printed in Italy

1^a edizione: settembre 2007

Sommario

Parte I Aspetti generali

1	Introduzione	3
1.1	Caratteristiche di un sistema embedded	5
1.2	Progettazione dei sistemi embedded	13
1.3	Evoluzione	16
1.3.1	Influenza dell'industria	17
1.3.2	Electronic design automation	20
1.3.3	Piattaforme di sviluppo	21
1.4	Il mercato dei sistemi embedded	23
1.4.1	Software	25
1.4.2	Hardware	27
1.5	Riepilogo	30

Parte II Realizzazione dei sistemi

2	Architettura e progettazione	35
2.1	Printed Circuit Board	37
2.1.1	Componenti	37
2.1.2	Supporto	39
2.1.3	Montaggio dei componenti	44
2.1.4	Approccio alla progettazione	45
2.2	System-on-Chip	47
2.2.1	Approccio alla progettazione	49
2.2.2	Design for testability	51
2.3	Sistemi distribuiti	57
2.3.1	Alcuni ambiti applicativi	57
2.3.2	Wireless sensor network	59
2.3.3	Approccio alla progettazione	61

2.4	Piattaforme di prototipazione	63
2.4.1	Tipologie di sistemi di sviluppo	64
2.4.2	Approccio alla progettazione	65
2.4.3	Esempi	67
2.5	Riepilogo	72
3	Flusso di progettazione	75
3.1	Introduzione	75
3.1.1	Modelli	76
3.1.2	Metriche	83
3.1.3	Strumenti	86
3.2	Flusso di sviluppo a livello di sistema	94
3.2.1	Aspetti generali	94
3.2.2	Alcuni flussi specifici	96
3.3	Flusso di sviluppo di PCB	99
3.4	Flusso di sviluppo dell'hardware	101
3.4.1	Front-end	102
3.4.2	Back-end	106
3.5	Flusso di sviluppo del software	109
3.5.1	Flusso di sviluppo di sistemi operativi	110
3.5.2	Flusso di sviluppo di software applicativo	112
3.6	Riepilogo	116
4	Pianificazione e gestione	119
4.1	Ciclo di sviluppo	120
4.2	Modelli	122
4.2.1	Cascata	123
4.2.2	Ciclo a V	124
4.2.3	Spirale	124
4.2.4	Contratto	125
4.2.5	Una visione più realistica	126
4.3	Project management	127
4.3.1	Problemi di gestione	128
4.3.2	Organizzazione	129
4.4	Costi di sviluppo	133
4.4.1	Costo di sviluppo dell'hardware	134
4.4.2	Costi di sviluppo del software	138
4.4.3	Costi indotti dal mercato	146
4.5	Riuso e proprietà intellettuali	148
4.5.1	Productivity gap	148
4.5.2	Metodologia ed economia di progetto	151
4.5.3	Hardware libero	155
4.6	Creazione di prodotti	156
4.6.1	Gerarchie e ruoli nella progettazione	156
4.6.2	Gestione del rischio	157
4.7	Riepilogo	159

Parte III Elaborazione

5	Tecnologie hardware	163
5.1	Tecnologie ASIC	164
5.1.1	Processo planare	164
5.1.2	Standard cell	169
5.1.3	Gate array	172
5.1.4	Full custom	173
5.2	Tecnologie programmabili	173
5.2.1	Classificazione	173
5.2.2	PLA, PAL e GAL	176
5.2.3	CPLD	178
5.2.4	FPGA	178
5.3	Riepilogo	179
6	Microprocessori	181
6.1	Microprocessori general purpose	185
6.1.1	Architetture CISC	186
6.1.2	Architetture superscalari	189
6.1.3	Architetture CISC/RISC	190
6.1.4	Architetture RISC	191
6.1.5	Architetture EPIC/VLIW	193
6.1.6	Analisi comparata	195
6.2	Processori dedicati	195
6.2.1	Digital Signal Processor	196
6.2.2	Network processor	200
6.2.3	Microcontrollori	202
6.3	Riepilogo	204
7	Memorie	207
7.1	Classificazione e parametri caratteristici	208
7.2	Memorie non volatili	211
7.2.1	ROM	211
7.2.2	EPROM, E ² PROM	213
7.2.3	Flash	215
7.2.4	Memorie di massa embedded	218
7.3	Memorie volatili	219
7.3.1	Organizzazione	219
7.3.2	Memorie statiche	220
7.3.3	Memorie dinamiche	222
7.4	Gerarchia di memoria	223
7.5	Prestazioni	232
7.6	Riepilogo	234

8	Software di base	235
8.1	Sistemi operativi	236
8.1.1	Caratteristiche generali	236
8.1.2	Sincronizzazione e comunicazione	240
8.2	Gestione del processore	243
8.2.1	Stati di un processo	243
8.2.2	Criteri di scheduling	245
8.3	Esigenze dei sistemi real-time	249
8.4	Scheduling per il real-time	251
8.4.1	Tempi di risposta	251
8.4.2	Classi di algoritmi	255
8.4.3	RMS, EDF e quote proporzionali	256
8.5	Real-Time Operating System	258
8.5.1	Linux/UNIX	260
8.5.2	Windows	262
8.5.3	VxWorks	263
8.5.4	Windows CE	265
8.5.5	Free software	269
8.5.6	Patch real-time	271
8.6	Riepilogo	275
9	Software applicativo	277
9.1	Scelta del linguaggio	277
9.2	Qualità del codice	280
9.2.1	Stile di scrittura	282
9.2.2	Analisi e documentazione	287
9.2.3	Metriche	290
9.3	Validazione	293
9.3.1	Testing del software	294
9.3.2	Verifica del sistema embedded	296
9.3.3	Software per uso domestico	301
9.4	Riepilogo	306

Parte IV Comunicazione

10	Segnali	309
10.1	Conversione analogico/digitale	311
10.1.1	Campionamento	311
10.1.2	Quantizzazione	312
10.1.3	Dithering	314
10.1.4	Convertitori A/D	315
10.2	Conversione digitale/analogico	321
10.2.1	Convertitori D/A	324
10.3	Rappresentazione fisica dei livelli logici	327
10.4	Esempio applicativo	329

10.5	Riepilogo	332
11	Comunicazione	333
11.1	Trasmissione seriale	333
11.1.1	Trasmissione seriale sincrona	334
11.1.2	Trasmissione seriale asincrona	337
11.1.3	Protocolli e bus seriali standard	338
11.2	Trasmissione parallela	342
11.2.1	Meccanismi di comunicazione di base	343
11.2.2	Protocolli paralleli standard	345
11.3	Trasmissione wireless	346
11.3.1	Modulazione analogica	346
11.3.2	Modulazione digitale	349
11.4	Codici a rilevamento e correzione di errore	353
11.4.1	Codici a ripetizione	354
11.4.2	Codici di Hamming	355
11.4.3	Codici a ridondanza ciclica	357
11.5	Riepilogo	358
12	Interfacciamento	361
12.1	Indirizzamento	362
12.1.1	Memory mapped I/O	362
12.1.2	Standard I/O	364
12.1.3	Port mapped I/O	364
12.1.4	Extended I/O	366
12.2	Polling	367
12.3	Interrupt	368
12.3.1	Architetture per la gestione di interrupt e arbitraggio	370
12.3.2	Identificazione della periferica	375
12.3.3	Mascheramento	376
12.4	Accesso diretto alla memoria	378
12.5	Riepilogo	380
13	Sensori	383
13.1	Caratteristiche	385
13.2	Esempi	388
13.2.1	Posizione	388
13.2.2	Velocità	389
13.2.3	Temperatura	389
13.2.4	Tensione meccanica	391
13.2.5	Umidità	391
13.2.6	Luminosità	391
13.2.7	MEMS	392
13.3	Riepilogo	392

Appendici

A	Studio di fattibilità di single-chip	397
A.1	Analisi del progetto	399
A.2	Soluzione ASIC	400
A.3	Flusso di progettazione	402
A.4	Selezione dei fornitori	404
A.5	Tradeoff	406
A.6	Considerazioni conclusive	410
A.7	Riepilogo	413
B	Linguaggi di descrizione dell'hardware	415
B.1	Introduzione	415
B.2	VHDL	416
B.2.1	Livelli di astrazione	416
B.2.2	Design entities	417
B.2.3	Statement	419
B.2.4	Pipelining	427
B.2.5	Librerie IEEE	428
B.3	Verilog	431
B.3.1	Livelli di astrazione	431
B.3.2	Costanti	432
B.3.3	Segnali	433
B.3.4	Operatori	435
B.3.5	Moduli	437
B.3.6	Blocchi dichiarativi	442
B.3.7	Blocchi procedurali	448
B.4	Riepilogo	455
	Glossario	457
	Bibliografia	473
	Indice analitico	479

Prefazione

Alzarsi la mattina, fare colazione e recarsi con calma in libreria per acquistare questo testo. È sabato o domenica, finalmente un giorno in cui non si interagisce con la tecnologia, in cui non si accende il computer.

Tale sogno non è prossimo alla realtà. Da qualche ora si è probabilmente già entrati in contatto con un centinaio di sistemi basati su microprocessore: decine sull'automobile, due o tre in ogni elettrodomestico o telefono cellulare, almeno uno in ogni apparato alimentato dalla rete elettrica o a pile: condizionatore, apricancello, pace-maker, calzature ginniche *evolute*, semafori per l'attraversamento pedonale, carte di credito per il pagamento elettronico, e così via.

La maggior parte dei sistemi di elaborazione non è costituita da personal computer, bensì da dispositivi in stretta relazione con l'ambiente in cui operano, che in genere hanno uno scopo prefissato e per tale motivo non richiedono di caricare programmi o, spesso, di avere interfacce con tastiera e monitor. Si parla di sistemi *invisibili*, *ubiqui* e *pervasivi*, per rappresentare la loro trasparente invasività nella vita quotidiana. Questi sistemi, detti *dedicati* o *embedded*, dominano pertanto numericamente il mercato, anche se in modo non conclamato, rispetto ai sistemi *general purpose* come il PC.

Perché un libro sui sistemi embedded?

La loro architettura ricorda quella di un generico sistema di calcolo, con sezioni di elaborazione, comunicazione, memorizzazione e interfaccia, ma i requisiti operativi li rendono talmente peculiari da richiedere metodologie di progetto, e un approccio alla ricerca di soluzioni tecnologiche, tali da giustificare la nascita di una vera e propria nuova disciplina ingegneristica.

Fra le caratteristiche distintive compare la necessità, dati i segmenti di mercato in genere di larga diffusione, di contenere il più possibile i costi del prodotto e i tempi di realizzazione. I sistemi sono fortemente specializzati e ottimizzati per svolgere un ristretto numero di compiti, con vincoli stringenti su consumo di potenza, prestazioni, dimensioni, affidabilità e così via. Le principali conseguenze di un così elevato numero di requisiti sono: la scelta di architetture miste hardware/software per bilanciare i

costi, le prestazioni e i tempi di sviluppo; il ricorso estensivo a strumenti CAD per supportare la simulazione e la sintesi dei sistemi; la messa a punto di metodologie di progetto che iniziano considerando l'applicazione a livello *sistema*, traducendo in seguito tali specifiche in un dispositivo reale sulla base dei vincoli e degli obiettivi, senza ricalcare necessariamente la struttura general purpose di un PC.

Per tali ragioni il progetto dei sistemi embedded non è convenzionale. Si devono avere presenti le opportunità offerte da ogni possibile tecnologia realizzativa, sia essa hardware o software, e adottare un approccio al design che ottimizzi in modo concorrente tutti i requisiti del progetto.

L'obiettivo del libro è fornire le basi metodologiche per applicare efficacemente un approccio *trasversale* nelle tecnologie e *olistico* nel risultato: progettare in modo flessibile sistemi dedicati a svolgere in modo ottimizzato compiti con un elevato grado di specializzazione.

A chi è rivolto il libro

Il libro è rivolto a studenti universitari e professionisti che si trovano a dover progettare e/o utilizzare sistemi, i cui requisiti funzionali e i vincoli di progetto portano a utilizzare architetture di calcolo specializzate. Si suppone che il lettore abbia conoscenze, almeno di base, relative alla struttura di un sistema di calcolo, al ruolo o all'organizzazione del sistema operativo e possieda i rudimenti della programmazione. La scelta degli argomenti di ciascuno dei capitoli riflette l'obiettivo di fornire una chiave di lettura per ogni aspetto della progettazione e analisi dei sistemi embedded, che racchiude sia gli aspetti di design dell'hardware e software, sia la gestione del progetto. Ogni capitolo è stato concepito per essere il più possibile autocontenuto, in modo che il lettore, e in particolare il docente, possa facilmente personalizzare il percorso di approfondimento delle tematiche.

Si è cercato di trasferire soprattutto l'aspetto *metodologico* nell'analisi dei problemi, per consentire di valutare e selezionare ogni novità tecnologica, cogliendone le eventuali opportunità ma, nel contempo, evitando che l'enfasi sulle soluzioni *dell'ultima ora* portino lontano dagli obiettivi (e vincoli) reali del progetto.

Struttura del volume

L'organizzazione del testo prevede quattro parti che ricalcano gli aspetti rilevanti dell'analisi e progettazione dei sistemi.

La *prima parte* conduce il lettore nel mondo dei sistemi dedicati.

Nel Capitolo 1 si evidenziano le peculiarità che distinguono i sistemi embedded dai sistemi general purpose, descrivendo come gli obiettivi e vincoli di progetto si riflettono sulle scelte architetture. Vengono infine evidenziate e caratterizzate le forze che spingono l'evoluzione dei sistemi embedded, ovvero i *trend* del mercato e i *progressi* tecnologici attesi.

Nella *seconda parte* l'attenzione viene posta sulle *metodologie* di progettazione.

Il Capitolo 2 introduce soluzioni, tecnologie disponibili e metodologie per la realizzazione di sistemi misti hardware/software per applicazioni embedded. L'obiettivo è fornire una panoramica degli ambiti applicativi, dei pregi e dei limiti delle varie soluzioni presenti sul mercato: si spazia dal tradizionale (e longevo) montaggio su board,

per arrivare alla realizzazione dell'intero sistema su un unico circuito integrato. Per le varie tecnologie sono esaminati i pregi e le limitazioni che possono per esempio essere indotte dall'elevato livello d'integrazione, come il testing delle funzionalità.

Il Capitolo 3 riguarda i vari passi che consentono di tradurre le funzionalità astratte di un sistema in un'architettura reale. Dopo una fase di analisi delle specifiche, per buona parte dei sistemi embedded il problema principale è selezionare l'architettura su cui *mappare* le funzionalità, procedendo quindi alla sintesi di tutti i componenti: hardware, software e interfacce di comunicazione. Il capitolo offre una completa panoramica del flusso di progettazione a livello di sistema, specializzando in seguito i percorsi per il software, l'hardware e lo sviluppo della board. Particolare attenzione è stata rivolta nell'evidenziare l'importanza di comporre una toolchain di strumenti di sviluppo, a supporto di un realistico flusso di progettazione del sistema.

Il Capitolo 4 conclude la seconda parte del volume, richiamando l'attenzione su aspetti talvolta trascurati: la pianificazione e la gestione delle attività di sviluppo. Il successo e la progettazione di un sistema embedded, infatti, non dipendono solo da fattori tecnologici, ma sono anche espressione della capacità di organizzare il lavoro e di sapere prendere le decisioni migliori all'inizio del processo di sviluppo, quando il livello d'incertezza è ancora elevato. A tal fine sono presentati i più diffusi approcci organizzativi, illustrati i principali modelli per la previsione dei tempi e costi e chiarito il ruolo e le responsabilità di un *project manager* nella realizzazione di un prodotto per il mercato dei sistemi embedded. Particolare attenzione viene rivolta alla comprensione delle implicazioni della progettazione basata sul riuso.

La *terza parte* è dedicata alla presentazione degli aspetti di *elaborazione*.

Le tecnologie e i dispositivi per realizzare le sezioni di calcolo sono il contenuto dei Capitoli 5, 6 e 7, che presentano gli approcci per implementare i sistemi digitali dedicati e le diverse architetture di microprocessore.

La trattazione si articola partendo da soluzioni puramente hardware, con elevato grado di specializzazione. Il Capitolo 5 si concentra sugli approcci che sfruttano la tecnologia dei circuiti integrati, ottimizzando il livello d'integrazione (ASIC) o la flessibilità (logiche programmabili). I processori, descritti nel Capitolo 6, vengono classificati sulla base del loro grado di specializzazione, dettagliando alcune soluzioni dedicate, come i microcontrollori e i DSP, utilizzate efficacemente in molti sistemi embedded.

Il sottosistema di memoria è l'oggetto del Capitolo 7 dove si riportano le caratteristiche delle principali tecnologie allo stato solido. Si analizzano in maniera comparata le possibili scelte, in funzione dei requisiti dell'applicazione e delle peculiarità delle varie tecnologie. Viene inoltre descritta l'organizzazione di una gerarchia di memoria, evidenziandone i pregi, ma anche i limiti di complessità che la rendono inadatta per un numero significativo di applicazioni embedded.

Il Capitolo 8 è dedicato a inquadrare il ruolo del software di base e le peculiarità che deve assumere in relazione ai vincoli di reattività e real-time dei sistemi embedded. Dopo un richiamo al ruolo e organizzazione del sistema operativo, si analizzano le esigenze dei sistemi real-time in termini di latenza nella risposta agli eventi e determinismo nel comportamento. Si sottolinea come tali esigenze si ripercuotano sulla struttura dei sistemi operativi real-time, soprattutto per quanto concerne la strategia di gestione e scheduling della CPU. A chiusura del capitolo sono analizzati alcuni

sistemi operativi reali, per comprendere se e quando, in base alla loro complessità e organizzazione, possono essere un valido supporto alla realizzazione di un sistema embedded.

La visione *applicativa* del software è contenuta nel Capitolo 9, dove si considerano in modo congiunto gli aspetti legati all'architettura di calcolo, il processo di sviluppo e gli ambienti di supporto allo scrittura del codice. Si descrivono le caratteristiche salienti di un linguaggio di programmazione per applicazioni embedded, presentando al lettore gli strumenti per valutare e migliorare diversi aspetti del software: prestazioni, chiarezza stilistica, documentazione, efficienza e correttezza. L'obiettivo è trasferire al lettore l'esigenza e l'importanza di scrivere codice di *qualità*, come primo passo verso la correttezza e la robustezza dell'applicazione complessiva. Il problema della verifica viene affrontato in maniera incrementale, partendo dalle problematiche di testing a livello del codice, per giungere alla simulazione del comportamento a livello di sistema. Il capitolo si conclude con un caso di studio che presenta le ripercussioni sul software associate al processo di certificazione di qualità dei prodotti per uso domestico.

La *quarta parte* è dedicata alla *comunicazione* e all'*interfacciamento* fra i sistemi.

Il Capitolo 10 introduce il concetto di segnale e tratta le problematiche di conversione analogico/digitale e viceversa. Tale problema è fondamentale poiché, sebbene in un sistema embedded i calcoli sono svolti principalmente nel dominio digitale, l'interazione con l'ambiente è forte, con la conseguente esigenza di rilevare e controllare grandezze analogiche. La necessità d'interfaccia fra il dominio analogico e digitale è quindi cruciale, e il capitolo presenta le principali architetture evidenziando limiti e pregi di ogni soluzione.

Nel Capitolo 11 si prende in considerazione come trasferire le informazioni da una sezione all'altra di un sistema, ovvero la comunicazione. Le strategie presentate vengono inquadrare in termini di requisiti e struttura della soluzione, in modo da caratterizzarle in base alla distanza coperta, natura e velocità del canale di trasmissione. La trattazione si articola dalle soluzioni più semplici per la trasmissione seriale di informazioni, per giungere ad affrontare le problematiche di trasmissione delle informazioni con tecnologie wireless, esplorando nel contempo il tema della codifica delle informazioni per una maggiore immunità dai disturbi.

L'interfacciamento, descritto affrontando gli aspetti d'*indirizzamento* e *sincronizzazione*, è il tema del Capitolo 12. Tale chiave di lettura del problema viene utilizzata per i diversi ambiti applicativi. Si presentano dapprima le strategie d'interfacciamento tra dispositivi hardware e fra gli elementi di un sistema basato su microprocessore, descrivendo architetture di base, politiche di gestione, aspetti legati al software e, infine, limitazioni di ogni approccio.

Il tema dell'interfacciamento si conclude nel Capitolo 13 che affronta il problema di acquisire informazioni dal mondo reale, tramite sensori. Si presentano i principali elementi d'interesse per un progettista: come cogliere gli scostamenti dal comportamento ideale e quali sono le principali famiglie di sensori disponibili.

A chiusura del libro sono riportate due *appendici*.

La prima è un caso di studio che affronta una classica situazione in cui, in tempi stretti, si devono coniugare le competenze tecniche con gli aspetti di gestione della progettazione. Si tratta di uno studio di fattibilità per valutare la convenienza nell'adottare una soluzione single-chip per l'aggiornamento e l'evoluzione di un sistema già

esistente, basato sull'uso di componenti discreti e FPGA. Sono contenute molte tematiche che si affrontano nella *realtà*, come la selezione e l'interazione con una foundry di silicio e il confronto economico fra diverse tecnologie.

La seconda appendice inizia presentando il ruolo dei formalismi per la specifica dell'hardware all'interno di un flusso di progettazione.

Il prosieguito costituisce una valida guida introduttiva per comprendere che cosa significhi rappresentare un'architettura di calcolo reale tramite un linguaggio di descrizione dell'hardware. Con riferimento ai due formalismi più diffusi (VHDL e Verilog), sono discussi i vari passi che portano dal livello più astratto della simulazione funzionale del comportamento, sino alla sintesi e verifica dell'hardware finale.

Chiudono il volume un dettagliato glossario e una estesa bibliografia di riferimento.

Carlo Brandolese e William Fornaciari

Politecnico di Milano
Dipartimento di Elettronica e Informazione