Indice

Prefazione **XI**

Introduzione **XIII**

1 Comportamento a stati finiti di un sistema embedded **3**
1.1 Richiami su automi a stati finiti riconoscitori di linguaggi . . . 4
1.2 Grammatiche . . . 5
1.3 Linguaggi non regolari . . . 6
1.4 Classificazione di Chomsky . . . 7
1.5 Macchine a stati finiti . . . 9
1.6 Implementazione di automi a stati finiti . . . 11
1.7 Implementazione di macchine a stati finiti . . . 13
1.8 Event-driven systems . . . 14

2 Sistemi in tempo reale **17**
2.1 Temporizzazione dell'algoritmo di controllo . . . 18
2.2 Multitasking . . . 21
2.2.1 Scheduling statico . . . 24
2.2.2 Scheduling dinamico . . . 26
2.2.3 Test di scheduling basati sull'utilizzo . . . 28
2.2.4 Interazione tra i task . . . 31
2.2.5 Protocolli priority ceiling . . . 33
2.3 Sistemi operativi real time . . . 34

3 Introduzione ai processori di utilizzo industriale **37**
3.1 MCU . . . 39

3.2 MPU 44
3.3 DSP 44
3.4 PLC 46
3.5 PC industriali 49
3.6 Sviluppo host-target 49
3.7 Le smartcard 50

4 I concetti della Dependability 57
4.1 Attributi della dependability 58
4.2 Impedimenti alla dependability 59
4.2.1 Classificazione dei guasti 60
4.2.2 Errori 61
4.2.3 Fallimenti 62
4.3 Mezzi per ottenere dependability 64

5 Valutazione quantitativa degli attributi di dependability 67
5.1 Affidabilità 67
5.2 MIL-HDBK 217F 70
5.3 Safety 73
5.4 Manutenibilità 74
5.5 Disponibilità 74
5.6 Valutazione dell'affidabilità di un sistema 76
5.6.1 Il metodo combinatorio 76
5.6.2 Metodo enumerativo 79
5.6.3 Modello di Markov 81
5.7 Valutazione markoviana di altri attributi 84
5.7.1 Safety 84
5.7.2 Disponibilità 85
5.8 Considerazioni conclusive 86

6 Valutazione qualitativa degli attributi di safety 89
6.1 Il concetto di rischio 89
6.2 Failure Mode and Effect Analysis (FMEA) 94
6.3 Hazard and Operability study (HAZOP) 95
6.4 Fault Tree Analysis 97
6.5 Analisi probabilistica di un Fault Tree 98

7 Tecniche di rilevazione dei guasti 103
7.1 Duplicazione e confronto 104
7.2 Duplicazione complementata 105
7.3 Test di accettazione 106
7.4 Test diagnostici 107
7.5 Rilevazione temporale dell'errore 108
7.6 Monitoraggio del processore 109

8 Codici rilevatori di errore 111
8.1 Codici di parità 113

8.2 Checksum 115
8.3 Codice m-su-n 116
8.4 Duplicazione 117
8.5 Codici ciclici 118
8.5.1 Errori singoli 122
8.5.2 Errori dispari 123
8.5.3 Errori doppi 123
8.5.4 Errori a burst 124
8.5.5 Utilizzo dei codici ciclici 124
8.6 Codici aritmetici 125

9 Codici correttori di errore **129**
9.1 Overlapped parity 129
9.2 Codici di Hamming 131
9.3 Codici correttori aritmetici 134
9.4 Codici CRC correttori 135
9.5 Codici Reed Solomon 136
9.5.1 Gruppi e Campi di Galois 136
9.5.2 Il campo esteso GF(2^m) 138
9.5.3 L'operazione di somma nel campo esteso GF(2^m) 138
9.5.4 I polinomi primitivi 139
9.5.5 Il campo esteso GF(2^3) 140
9.5.6 La codifica Reed-Solomon 142
9.5.7 La codifica sistematica tramite shift register 145
9.5.8 La decodifica 146
9.5.9 Applicazioni dei codici Reed Solomon 149

10 Tolleranza ai guasti **151**
10.1 Error detection and recovery 152
10.2 Duplicazione Riconfigurabile 153
10.3 Fault Masking 155
10.3.1 N-Modular Redundancy 155
10.3.2 Meccanismi di votazione 156
10.4 NMR Riconfigurabile 158
10.5 Shadow box 161

11 Algoritmi distribuiti **163**
11.1 Memoria stabile 164
11.2 Checkpointing in ambito distribuito 166
11.3 Two-Phase Commit Protocol 168
11.4 Paradosso dei generali bizantini 169
11.5 Consenso Distribuito tra processi asincroni 170
11.5.1 L'algoritmo di Chandra e Toueg 171
11.5.2 Primitive di comunicazione 172
11.5.3 Primitive del consenso 173
11.6 Byzantine Agreement 176

11.7 Interactive Consistency 179
11.8 Algoritmi di Sincronizzazione di Clock in ambito distribuito . . 179
11.8.1 Algoritmi probabilistici 180
11.8.2 Algoritmi di consistenza interattiva 181
11.8.3 Algoritmi a convergenza: algoritmi a media 181
11.8.4 Algoritmi a convergenza: algoritmi non a media 181

12 I guasti software **185**
12.1 Il caso dell'Ariane 5 186
12.2 Diversità 188
12.2.1 Recovery block 190
12.2.2 N-Version Programming 191
12.2.3 Esempi di uso della diversità in sistemi avionici 191
12.3 Programmazione Difensiva 192
12.4 Affidabilità del software 193
12.5 Standard di codifica 196
12.6 Sottoinsiemi di linguaggi 197
12.6.1 SafeAda 198
12.6.2 MISRA C 199
12.6.3 Javacard 200
12.7 Validazione dei compilatori 200

13 Verifica del codice **203**
13.1 Definizioni relative all'attività di testing 204
13.2 Test di unità 205
13.2.1 Testing funzionale 206
13.2.2 Test strutturale e criteri di copertura 207
13.2.3 Test statistico 212
13.2.4 Conduzione del test di unità 213
13.2.5 Object-Oriented Testing 215
13.3 Test di integrazione 215
13.4 Test di sistema 216
13.5 Test di accettazione 217
13.6 Test di regressione 217
13.7 Analisi mutazionale 218
13.8 Strumenti di supporto al testing 218
13.9 Analisi statica 219
13.9.1 Interpretazione astratta 220

14 Metodi Formali **223**
14.1 Verifica formale del codice sorgente 224
14.2 I metodi asserzionali: il metodo B 225
14.3 Le logiche classiche 231
14.4 Logica modale 234
14.5 Logica temporale 236
14.5.1 LTL 237

14.5.2 CTL 239
14.6 L'algoritmo di Model Checking 241
14.7 Model Checking simbolico 245
14.7.1 Binary Decision Diagrams 245
14.7.2 Rappresentazione dello spazio degli stati con BDD . . . 249
14.7.3 Algoritmo di Symbolic Model Checking 250
14.7.4 Utilizzo del model checking simbolico 252
14.8 Model Driven Development 253

15 La certificazione del software **261**
15.1 Certificazione di processo e di prodotto 262
15.2 Principali normative sulla safety dei sistemi safety-critical . . . 263
15.2.1 Settore militare 263
15.2.2 Settore avionico/spaziale 264
15.2.3 Settore ferroviario 264
15.3 Ciclo di vita del software 264
15.4 Safety Integrity level 265
15.5 Esempi di tecniche richieste dalle normative per il software . . 267

Bibliografia **271**