

Contents

	Preface	ix
	Acknowledgements	xi
1	Why VMEbus ?	1
	Why use a bus at all?	1
	The introduction of the MC68000	3
	New bus requirements	12
	VERSAbus — the precursor.	12
	VMEbus	13
	But what does VME stand for ?	13
2	VMEbus Primer	14
	VMEbus: its design concepts	14
	The data transfer bus	15
	The DTB arbitration bus	26
	Interrupt handling	30
	The utility bus	33
	VMEbus revision D extensions	34
	Future developments	37
	Mechanical specifications	38
3	VSB and secondary buses	39
	VMX — the first subsystem bus	39
	VSB — the VMEbus subsystem bus:	40
	VMSbus serial bus interface	49
	VXI — VMEbus extensions for instrumentation	50
	VICbus — vertical intercrate bus	52
4	VMEbus I/O systems	55
	Physical connections	55
	Intelligent and non-intelligent boards	56

Using a dedicated I/O bus	61
Modern I/O buses	62
Connecting I/O buses to VMEbus	70
Onboard I/O extension buses	70

5 Operating systems 76

What do operating systems provide ?	76
Single tasking operating systems	81
Multitasking operating systems	83
What is a real-time operating system ?	90
Real-time operating systems and UNIX	93
pSOS+	98
OS-9	101
VXWorks	102
VRTX-32	103
LynxOS — POSIX conformance	104
VMEPROM	106
VMEexec	107
Choosing an operating system	108

6 Single processor systems 114

Controllers and MASTERS	114
Anatomy of a processor card	114
Slave VMEbus card	118
Error handling	121
Printer buffer	122
The future for single processor systems	125

7 Multiprocessor techniques 127

Closely coupled processors	127
Loosely coupled multiprocessing systems	129
Sending messages	130
Message passing performance	131
Preventing message corruption	134
Multiprocessor architectures	139
Multiprocessing software	144
Scalable or superscalar systems	147

8	Multiprocessing with VMEbus	153
	Basic concepts	153
	A global message passing bus	154
	Mailbox locations	158
	Extending local resource	165
	Cache memory systems	166
	VME subsystem bus	169
	Synchronising the system	171
	Handling errors	172
	Protecting memory resources	174
	Multiprocessor organisations	178
9	Writing software for VMEbus systems	180
	The compilation process	181
	Using a bare compiler	185
	Run time libraries	185
	Writing a library	187
	Using alternative libraries	195
	Using a standard library	196
	Porting kernels	197
	Downloading	203
	Debugging techniques	209
	Using the target as a development system	227
10	System integration	230
	What is involved ?	230
	Work your way up	230
	Porting application software	237
11	Fault finding	239
	What to do if something goes wrong ?	239
	What is involved with a memory access ?	239
	Hardware or software to blame ?	241
	Partitioning the system	243

12	A design methodology	252
	Design methodology	252
13	Design studies	261
	Benchmark example	261
	Creating software state machines	266
	Burglar alarm	278
	Simple production machine simulator	285
	A data processor	290
	A cable harness tester	295
	A UNIX based cable harness tester	298
14	Interfacing to VMEbus	303
	Buffering	303
	SLAVE interface	304
	Interfacing 8 bit peripherals	306
	Using silicon support chips	319
15	High integrity systems	321
	Basic system integrity	321
	Improving fault tolerance	326
	Detecting processor failures	331
	Military approved VMEbus boards	338
Appendices		
A	Common system errors	340
B	Bus and interface pin outs	356
C	Interface chips	360
D	Common VMEbus specification violations	362
	Index	364

