
List of Figures

Chapter 1

Figure 1-1:	Evolution of communication vs. computation performance.	6
-------------	--	---

Chapter 2

Figure 2-1:	Patterson and Hennessy's definition of computer architecture.	13
Figure 2-2:	IBM 360 designers' definition of computer architecture.	14
Figure 2-3:	Communication architecture as facet of computer architecture.	15
Figure 2-4:	Communication architecture layers.	16
Figure 2-5:	LogP components of a parallel machine.	20

Chapter 3

Figure 3-1:	NCUBE 6400 processor block diagram.....	25
Figure 3-2:	NCUBE 6400 instruction set summary.....	27
Figure 3-3:	nCUBE/2 network interface organization.	28
Figure 3-4:	nCUBE/2 DMA instructions and registers.	30
Figure 3-5:	nCUBE/2 network timing.....	31
Figure 3-6:	CM-5 network backplane.....	33
Figure 3-7:	CM-5 processing node organization.	34
Figure 3-8:	CM-5 network interface access timing.....	35
Figure 3-9:	CM-5 fat tree data network topology.....	36
Figure 3-10:	CM-5 network interface organization.....	38
Figure 3-11:	CM-5 network interface status registers.....	38

Chapter 4

Figure 4-1:	Influences on Active Messages communication architectures.	44
Figure 4-2:	Active Messages model	46
Figure 4-3:	Components of a fetch&add implementation.	49
Figure 4-4:	Steps involved in sending and handling an Active Message.....	52
Figure 4-5:	nCUBE/2 Active Messages buffer management scheme.....	55
Figure 4-6:	Fetch&add implementation with nCUBE/2 Active Messages.....	60
Figure 4-7:	Send system call implementation for nCUBE/2 Active Messages.....	63
Figure 4-8:	Input DMA interrupt handler for nCUBE/2 Active Messages.....	65
Figure 4-9:	User-level interface to interrupt handler in nCUBE/2 Active Messages.....	67
Figure 4-10:	Matrices layout in blocks of columns for matrix multiply.	68

Figure 4-11:	Matrix multiply inner loop.	69
Figure 4-12:	Timing diagram for outer matrix multiply loop.	70
Figure 4-13:	Matrix multiply performance in Mflops.	71
Figure 4-14:	Matrix multiply processor and network utilization.	72
Figure 4-15:	Matrix transpose on 128 processors.	74
Figure 4-16:	Matrix transpose with constant message size.	75
Figure 4-17:	Usage of the Sparc register windows in CMAM.	82
Figure 4-18:	Fetch&add implementation with CM-5 Active Messages.	87
Figure 4-19:	Send CM-5 Active Message (CMAM_4) implementation.	90
Figure 4-20:	Send CM-5 Active Message (CMAM_4) timing.	91
Figure 4-21:	Poll CM-5 network for Active Messages.	91
Figure 4-22:	Dispatch on message tag for CM-5 Active Messages.	92
Figure 4-23:	Dispatch to CM-5 Active Message handler.	93
Figure 4-24:	An 16-input butterfly laid-out on four processors.	94
Figure 4-25:	Execution times for FFTs of various sizes on a 128 processor CM-5.	95
Figure 4-26:	FFT computation rates.	96
Figure 4-27:	FFT predicted and measured communication rates.	97
Figure 4-28:	Communication overhead shown graphically.	105

Chapter 5

Figure 5-1:	Blocking send&receive handshake protocol.	110
Figure 5-2:	Implementation of blocking send&receive on Active Messages.	111
Figure 5-3:	Pseudo-code for blocking send&receive using CM-5 Active Messages.	112
Figure 5-4:	Send&receive part 2.	113
Figure 5-5:	Timing for blocking send&receive implementation on CM-5 Active Messages.	113
Figure 5-6:	Timing estimate for send&receive implementation.	116
Figure 5-7:	Fortran-D compiled to non-blocking send and blocking receive.	119
Figure 5-8:	Monsoon processing element pipeline.	124
Figure 5-9:	Generic shared memory multiprocessor node structure.	128
Figure 5-10:	NUMA read and write using CM-5 Active Messages.	133
Figure 5-11:	Weak NUMA shared memory operations.	134

Chapter 6

Figure 6-1:	2D global address space underlying Split-C.	145
Figure 6-2:	Split-C pointer arithmetic.	147
Figure 6-3:	Software layers related to Split-C.	148
Figure 6-4:	Representation of global pointers in libsplit-c's 2-D address space.	149
Figure 6-5:	Usage of memory segments in Split-C.	150
Figure 6-6:	Address arithmetic in the two-dimensional address space.	151
Figure 6-7:	Split-phase remote memory accessors.	152
Figure 6-8:	Signalling stores.	153
Figure 6-9:	Predicted and measured performance for four Split-C sorting algorithms.	156
Figure 6-10:	TAM storage and scheduling hierarchies.	161
Figure 6-11:	TAM activation tree.	162
Figure 6-12:	Dynamic program execution statistics.	164

Chapter 7

Figure 7-1:	Typical code sequence for Split-C spread array indexing.	177
-------------	---	-----