Figure 4.10 Software synthesis for chirp signal generator. 102
Figure 4.11 Cosimulation 108
Figure 4.12 Block objects in Ptolemy send and receive data encapsulated in Particles to the outside world through Portholes. Buffering and transport is handled by the Geodesic and garbage collection by the Plasma. 110
Figure 4.13 The universal EventHorizon provides an interface between the external and internal Domains. 111
Figure 4.14 Hardware-software cosimulation using Ptolemy: system modeling 113
Figure 4.15 Hardware-software cosimulation using Ptolemy. 116
Figure 5.1 The design space. 119
Figure 5.2 User's view of the design flow and tools. 124
Figure 5.3 A design flow for the multiprocessor synthesis example 125
Figure 5.4 Internal representation of the tools and data. 127
Figure 5.5 Dependencies used for tool management. 128
Figure 5.6 Flow execution mechanisms. 129
Figure 5.7 The design flow for the multiprocessor synthesis example, specified within the DMM domain in Ptolemy 130
Figure 5.8 An example of Tool encapsulation: CodeGenerator tool. 131
Figure 5.9 Control panel in the DMM domain. 132
Figure 5.10 The Design Assistant implemented in the DMM domain 137
Figure 5.11 Run-time behavior of the Design Assistant with manual partitioning. 137
Figure 5.12 Hardware and software graphs generated by T3 138
Figure 5.13 Silage and CG56 assembly code generated by the hardware and software synthesis tools 139
Figure 5.14 Support for retargeting. 140
Figure 6.1 The Design Assistant 149
Figure A.1 Methodology for the estimation of hardware area required for a node when implemented at a sample period d. 183
Figure A.2 Illustration of possible nondeterminacy. 187

case 1: mapping based on GC only, case 2: threshold modified for repellers, case 3: complete local phase classification. 61

Figure 3.14 Algorithm trace: GC and the Mapping. All normal nodes, threshold = 0.5. 62

Figure 3.15 GC and mapping: (a) without extremity measures (b) with extremity measures. 63

Figure 3.16 GC and mapping: (a) without repeller measures (b) with repeller measures to change the threshold. 64

Figure 3.17 MIBS approach to solving extended partitioning 68

Figure 3.18 The bin selection procedure 70

Figure 3.19 Bin fraction and bin sensitivity for implementation-bin selection 74

Figure 3.20 Weighted Bin Sensitivity. 74

Figure 3.21 (a) Estimation of the hardware area for a node, for a given sample period (b) Computation of the hardware implementation curve for a node. 78

Figure 3.22 Comparison of MIBS and ILP solutions. 79

Figure 3.23 MIBS vs. GCLP (Extended Partitioning vs. Binary Partitioning) 80

Figure 3.24 (a) Node distribution among implementation bins. (b) Solution time of MIBS algorithm averaged over random graphs per graph size. 81

Figure 4.1 The target architecture. 85

Figure 4.2 Architecture of a hardware module. 87

Figure 4.3 The global controller for the hardware-software interface. 88

Figure 4.4 Cosynthesis approach....91

Figure 4.5 Hardware, software, and interface graphs. Each node in the hardware (software) graph contains a Silage or VHDL (C or assembly) description. 94

Figure 4.6 Mechanism for hardware synthesis for a single node....96

Figure 4.7 An example illustrating the hardware synthesis mechanism. 97

Figure 4.8 Architecture of the datapath and controller in a hardware module. This is the underlying architecture generated by Hyper, augmented with the ready and completion signals. 98

Figure 4.9 Mechanism for software synthesis from the software graph 100

LIST OF FIGURES

Figure 1.1 Task-level description of the receiver section of a modem. 1

Figure 1.2 System-level design 4

Figure 1.3 Hardware design options for a "node" at the algorithmic, transformational, and resource levels 5

Figure 1.4 Software design options for a node 7

Figure 1.5 Design specification hierarchy and translation across levels of abstraction.....8

Figure 2.1 The Design Assistant 13

Figure 2.2 A Codesign Flow 14

Figure 2.3 Models of computation used in system-level modeling 19

Figure 2.4 Typical target architectures for mixed hardware-software systems.....22

Figure 2.5 The target architecture 22

Figure 2.6 Specification semantics and target architecture.....23

Figure 3.1 The Extended Partitioning Problem 26

Figure 3.2 Typical hardware area-time trade-off curves, called hardware implementation curves. The design points are called implementation bins. 29

Figure 3.3 Implementation alternatives for a node within a mapping.....30

Figure 3.4 CH_i: Hardware implementation curve for node *i*. Each bin corresponds to a possible design option for the node. 31

Figure 3.5 Selection of the mapping objective at each step of GCLP. 39

Figure 3.6 The GCLP Algorithm. 40

Figure 3.7 Computation of Global Criticality 42

Figure 3.8 Hardware (EX_h) and software (EXs) extremity sets.....48

Figure 3.9 An example repeller property 50

Figure 3.10 Receiver section of a modem, described in Ptolemy. Hierarchical descriptions of the AGC and timing recovery blocks are shown.....57

Figure 3.11 Estimation of area and time values in hardware and software..... 57

Figure 3.12 GCLP vs. ILP: Random examples 59

Figure 3.13 Effect of local phase classification on GCLP performance