

Contents

1	Introduction	19
1.1	Motivation	19
1.2	Context	20
1.3	Definitions	21
1.4	Assumptions	26
1.5	Problem Analysis	27
1.6	Contributions	33
2	Related work	35
2.1	Hardware	36
2.1.1	Direct memory link	36
2.1.2	Multiple register banks processors/cores	38
2.1.3	Multithreading	40
2.1.4	Barrel processors	40
2.1.5	Caches	44
2.1.6	Locked/partitioned caches	47
2.1.7	Dual caches	48
2.1.8	Scratchpad memories	48
2.1.9	Specific virtual platform mapped to hardware	53
2.1.10	Specific hardware services	53
2.2	Software	57
2.2.1	Scheduling algorithms	57
2.2.2	Static analysis of code	58
2.3	Motivation, related to existing work	59
3	HwCS architecture	61
3.1	Improvement at L1 memory level	62
3.2	The HwCS	63
3.2.1	Predictability	67
3.2.2	Save/Restore ability	70
3.2.3	Layered usage	74
3.2.4	Benefits for real-time systems	75
3.3	Uniprocessor design	77
3.3.1	Architecture	77

CONTENTS

3.3.2	HwCS Real-time theory	77
3.3.3	Impact of the hardware context switch on the theory . . .	79
3.4	Multiprocessor design	84
3.4.1	Architecture	84
3.4.2	HwCS Real-time theory	84
3.5	Using the HwCS at L2 level	87
3.6	Example	88
3.7	The HwCS: beyond caches	91
3.8	Conclusion	91
4	Experimental setup and results	93
4.1	Implementation aspects	93
4.1.1	Degrees of freedom of the architecture	93
4.1.2	Hardware design	96
4.1.3	Area and timing considerations	99
4.2	Experimental setup	101
4.2.1	Operating System concepts	102
4.2.2	Benchmarks	102
4.2.3	Scheduling policy	103
4.2.4	Probes	104
4.3	Experiments	105
4.3.1	Simulation of experiments	107
4.3.2	Parameter extractions	112
4.3.3	Event processing	112
4.3.4	Memory tracking	112
4.4	Post-processing	112
4.5	Results	114
4.5.1	Short tasks	117
4.5.2	Medium tasks	121
4.5.3	Long tasks	124
4.5.4	The fantomatic very long task	124
4.5.5	Phenomenon affecting T20, T25 and T32: OS–task inter- ferences	127
4.5.6	Interferences, HwCS vs cache behaviour	131
4.5.7	About predictability	135
4.5.8	Conclusions	137
4.6	Conclusions on the experimental setup and results	141
5	Conclusion and further improvements	143
5.1	Further improvements	147
	References	162

A Appendix – Caches	163
A.1 Basic behaviour	163
A.2 Principle	164
A.3 Notion of locality	165
A.4 Disadvantages of caches	166
A.5 Access management	166
A.6 Associativity	167
A.7 Replacement policies	168
B Technological choices	172
C Software related events monitored	174
D System simulation profiling	176
E Softcore selection	178
F Uniprocessor design, task by task analysis	182
G Design hierarchy	254