

Light-Weight, Predictable Reactive Processing – The KEP Kiel Esterel Processor

Reactive Processing

Reactive Systems

- Control flow includes concurrency and preemption
- Real-time constraints
- Often low power budget

Reactive Systems on Traditional Processors

- Reactive control flow addressed at OS or compiler level
- Relatively Inefficient
- Difficult to predict

Reactive Processors

- Provide direct hardware support for reactive control flow
- Instruction Set Architecture (ISA) includes instructions for concurrency and preemption
- Strive for efficiency and predictability







Overview of KEP architecture, and detailed view of Reactive Block [2, 3]

The KEP Instruction Set Architecture

Closely matches Esterel language

- Synchronous model of computation
- Threads communicate through signals
- Time is divided into *ticks*

Inemonic, Operands	Esterel Syntax	Notes
AR <i>Prio, startAddr</i> [, <i>ID</i>] ARE <i>endAddr</i> DIN	[p q]	Fork and join, see also Section 5.1. An optional <i>ID</i> explicitly specifies the ID of the created thread.
RIO <i>Prio</i>		Set the priority of the current thread.
. T][W]ABORT [n,] S, endAddr	[weak] abort when [<i>n</i>] S	The prefix [L T] denotes the type of watcher to use, see also Section 5.2.
. T][W]ABORTI <i>Sexp</i> , endAddr	[weak] abort when immediate <i>Sexp</i>	T : Thread Watcher <i>none</i> : general Watcher
USPEND[I] Sexp, endAddr	suspend when [immediate] Sexp	
XIT addr	trap T in exit T end trap	Exit from a trap, <i>addr</i> specifies trap scope. Unlike GOTO, check for concurrent EXITs and terminate enclosing .
AUSE WAIT [n,] Sexp WAIT[I] Sexp	pauseawait [n] Sexpawait [immediate] Sexp	Wait for a signal. AWAIT TICK is equivalent to PAUSE.
AWAITS AWAIT[I] S, <i>addr</i> AWAITE	await case [immediate] <i>Sexp</i> do end	Wait for several signals in parallel.
IGNAL S	signal S in end	Initialize a local signal S.
$MIT \ S \ [, \ \{\#data reg\}]$	emit S [(val)]	Emit (valued) signal S.
USTAIN <i>S</i> [, {#data reg}]	sustain <i>S</i> [(<i>val</i>)]	Sustain (valued) signal S.
RESENT <i>S</i> , elseAddr	present S then end	Jump to <i>elseAddr</i> if S is absent.
OTHING	nothing	Do nothing.
ALT	halt	Halt the program.
OTO addr	loop end loop	Jump to <i>addr</i> .

An Esterel module illustrating the parallel and preemption statements (a), the translation rules for **every**, **sustain** and **loop** (b), and the resulting KEP3 assembler program (c) [2]

Concurrency in the KEP







Example code with false initialization of the Tick Manager (a), and resulting timing diagram (b) [1]

Experimental Results

Results for EstBench, compared to MicroBlaze 32bit Softcore

• Low power consumption:

Good scalability to high

degrees of concurrency:

	MicroBlaze		KEP3a		Ratio			
Module	(82mW@50MHz)		(mW)		(KEP to MB)		5)	
Name		Blank		Peak	Blank	Peak	Blan	k
abcd		69		13	8	0.16	0.12	2
abcdef		74		13	7	0.16	0.09)
eight_but		74		13	7	0.16	0.09)
chan_prot		70		28	12	0.34	0.17	1
reactor_ctrl		76		20	13	0.24	0.17	1
runner		78		14	2	0.17	0.03	;
example		77		25	9	0.30	0.12	2
ww_button		81		13	4	0.16	0.05	5
greycounter		78		44	33	0.54	0.42	2
Max. threads	2	10	20	40	60	80	100	120
Slices	1295	1566	1871	2369	3235	4035	4569	5233
Gates (k)	295	299	311	328	346	373	389	406

The KEP Evaluation Platform [3]

- Worst Case Reaction Time (**WCRT**) reduction between 50% and 85%
- Average Case Reaction Time (ACRT) reduction between 60% and 90%
- **Code Size** reduction between 25% and 90%

[1] Xin Li, Jan Lukoschus, Marian Boldt, Michael Harder, and Reinhard von Hanxleden. An Esterel Processor with Full Preemption Support and its Worst Case Reaction Time Analysis. In Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES), pages 225–236, New York, NY, USA, September 2005. ACM Press. [2] M Xin Li, Reinhard von Hanxleden. A concurrent reactive Esterel processor based onmulti-threading. In Proceedings of the 21st ACM Symposium on Applied Computing (SAC'06), Special Track Embedded Systems: Applications, Solutions, and Techniques, Dijon, France, April 23–27 2006. [3] Xin Li, Marian Boldt, Reinhard von Hanxleden. Mapping Esterel onto a Multi-Threaded Embedded Processor. In Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'06), San Jose, CA, October 21–25 2006. © Xin Li, R. v. Hanxleden 2007

Contact Person: Xin Li Department of Computer Science Christian-Albrechts-Universität zu Kiel Olshausenstr. 40, 24098 Kiel, Germany Phone: +49 (0)431 880-7290 Fax: +49 (0)431 880-7615 xli@informatik.uni-kiel.de http://www.informatik.uni-kiel.de/~xli

Contact Person: Prof. Dr. Reinhard von Hanxleden Department of Computer Science Christian-Albrechts-Universität zu Kiel Olshausenstr. 40, 24098 Kiel, Germany Phone: +49 (0)431 880-7281 Fax: +49 (0)431 880-7615 rvh@informatik.uni-kiel.de http://www.informatik.uni-kiel.de/~rvh **Further Information:** http://www.informatik.uni-kiel.de/rtsys/kep/