Reactive Processing
KReP, KEP, and Esterel

Malte Tiedje
Claus Traulsen

Real-Time Systems and Embedded Systems
Department of Computer Science
Christian-Albrechts-University Kiel

SLA++P 2008
Outline

KReP

HW Description with Esterel
KEP and KReP

Two reactive processors from Kiel:

KEP
- ISA based on Esterel
- Full support for concurrency and preemption
- Multi-threaded
- Implemented in VHDL and Esterel v7
- Available at
  www.informatik.uni-kiel.de/rtsys/kep

KReP
- Reactive processing without Esterel
- More dataflow oriented (Lustre, Scade)
- Multicore
- Implemented in Esterel v7 (prototype)
- Work in Progress
KEP and KReP

Two reactive processors from Kiel:

**KEP**
- ISA based on Esterel
- Full support for concurrency and preemption
- Multi-threaded
- Implemented in VHDL and Esterel v7
- Available at
  
  [www.informatik.uni-kiel.de/rtsys/kep](http://www.informatik.uni-kiel.de/rtsys/kep)

**KReP**
- Reactive processing without Esterel
- More dataflow oriented (Lustre, Scade)
- Multicore
- Implemented in Esterel v7 (prototype)
Two reactive processors from Kiel:

**KEP**
- ISA based on Esterel
- Full support for concurrency and preemption
- Multi-threaded
- Implemented in VHDL and Esterel v7
- Available at [www.informatik.uni-kiel.de/rtsys/kep](http://www.informatik.uni-kiel.de/rtsys/kep)

**KReP**
- Reactive processing without Esterel
- More dataflow oriented (Lustre, Scade)
- Multicore
- Implemented in Esterel v7 (prototype)
- Work in Progress
Processor: Basic Ideas

- Small Cores with simple ISA
- Special sync instruction
- Parallel or pipelined execution
- Fast and/or precise timing
- Special instructions for Automata
Compiler: Basic Ideas

Takes Lustre programs as inputs
1. Create dependency graph
2. Order Equations
3. add pre to preserve synchrony
4. Assign cores to variables

```lustre
node COUNTER(R:bool; I:bool)
    returns (C:int);
    T : int;
    E : int;
    let
        E = I and not pre(I);
        T = 0 -> if R
            then 0
            else if E
                then pre(T)+1
                else pre(T);
        C = T;
    tel
```
There exists at least one Lustre program, which can be compiled and executed on the KReP, giving the correct behavior.
Open Questions / To do

- Handle automata
- Benefit from Lustre clocks
- Timing Constraints
- Performance Evaluation
- Tune processor and compiler
- Correctness
Description of KEP and KReP

Both are described in Esterel v7

- Better maintainability
- Gain more experience with Esterel
Description of KEP and KReP

Both are described in Esterel v7

- Better maintainability
- Gain more experience with Esterel

It's also a nice Benchmark

**KEP**

- 6 Safe State Machines
- 76 Esterel Modules
- 4759 lines Esterel Code

**KReP**

- 3 Diagrams
- 26 Esterel Modules
- 1361 lines Esterel Code
Experience with Esterel v7

Mainly positive:

- Easy to use
- Nice formal tools
- SW + HW generation

Some minor problems:

- Weird compilation error (internal error in . . .)
- May not use complex expressions
- Code efficiency highly influenced by programming style
- State coverage not exhaustive

Two major ones:

- Cyclic programs
- Interfacing the real world
Experience with Esterel v7

Mainly positive:

▶ Easy to use
▶ Nice formal tools
▶ SW + HW generation

Some minor problems:

▶ Weird compilation error (internal error in . . .)
▶ May not use complex expressions
▶ Code efficiency highly influenced by programming style
▶ State coverage not exhaustive
Experience with Esterel v7

Mainly positive:
- Easy to use
- Nice formal tools
- SW + HW generation

Some minor problems:
- Weird compilation error (internal error in . . . )
- May not use complex expressions
- Code efficiency highly influenced by programming style
- State coverage not exhaustive

Two major ones:
- Cyclic programs
- Interfacing the real world
Cyclic Programs

- What is a cycle? Depends on the compiler.
- Hard to find (not local)
- Tool should show minimal cycle
- Main obstacles while programming
Interfaces: Synchronous view

- **tick**
- **rom_addr**
  - Shaded area labeled 1
  - Next tick, shaded area labeled 2
- **instr_from_rom**
  - Shaded area labeled D(1)
  - Next tick, shaded area labeled D(2)
Interfaces: Real world
Interfaces: Solution

- Add pause statements
- Request acknowledgement
- Multiclock
Conclusion

KReP

- Dataflow reactive processor seams feasible
- Still a lot to do
- Benefits?
Conclusion

KReP
- Dataflow reactive processor seems feasible
- Still a lot to do
- Benefits?

Esterel
- Easy to get prototype HW
- Hard to get efficient implementation
- Need some HW knowledge
- Interfaces and Cycles main problems