HW/SW Co-Design for a Reactive Processor

Sascha Gädtke, Xin Li, Marian Boldt, Reinhard von Hanxleden

Real-Time and Embedded Systems Group
Department of Comuter Science
Faculty Of Engineering
Christian-Albrechts-University Kiel, Germany

Student Poster Session, LCTES'06
Ottawa, June 14-16, 2006

Introduction

Classical approach: Run an Esterel program on a reactive processor like the *Kiel Esterel Processor* (*KEP*)

Drawback: Complex signal expressions have to be sequentialized into a series of KEP assembler instructions

Our Approach: Accelerate reactive processing via external logic block

```
module EXAMPLE:
input A, B, C;
output O, P;

present (A and C) or (B and C) then
  emit O
end;
present A or B then
  emit P
end
end module
```

```
      % module EXAMPLE
      INPUT A
      INPUT B
      INPUT C
      OUTPUT O
      OUTPUT P
      PRESENT A,A0
      PRESENT C,A0
      GOTO A1
  A0: PRESENT B,A2
      PRESENT C,A2
  A1: EMIT O
  A2: PRESENT A,A3
      GOTO A4
  A3: PRESENT B,A5
  A4: EMIT P
  A5: HALT
```

Figure: EXAMPLE Esterel program and resulting KEP assembler code

Approach

First Step: Source Code Transformation

Original Esterel program partitioned into three modules

Approach

First Step: Source Code Transformation

Original Esterel program partitioned into three modules

- SW module
  - Copy of the original program
  - Complex signal expressions are replaced with auxiliary signals

```
module EXAMPLE_SW:
output O;
output P;
input A_and_C_or_B_and_C;
input A_or_B;

present A_and_C_or_B_and_C then
  emit O
end present;
present A_or_B then
  emit P
end present

end module
```

Approach

First Step: Source Code Transformation

Original Esterel program partitioned into three modules

- SW module
  - Copy of the original program
  - Complex signal expressions are replaced with auxiliary signals
- HW module
  - Computes complex signal expressions in parallel threads
  - At any time a signal expression is TRUE, the auxiliary signal is emitted

```
module EXAMPLE_SW:
output O;
output P;
input A_and_C_or_B_and_C;
input A_or_B;

present A_and_C_or_B_and_C then
  emit O
end present;
present A_or_B then
  emit P
end present

end module
```

```
module EXAMPLE_HW:
input A, C, B;
output A_and_C_or_B_and_C, A_or_B;

  every immediate [A_or_B and C] do
    emit A_and_C_or_B_and_C
  end every
||
  every immediate [B or A] do
    emit A_or_B
  end every

end module
```

Approach

First Step: Source Code Transformation

Original Esterel program partitioned into three modules

- SW module
  - Copy of the original program
  - Complex signal expressions are replaced with auxiliary signals
- HW module
  - Computes complex signal expressions in parallel threads
  - At any time a signal expression is TRUE, the auxiliary signal is emitted
- Main module runs SW- and HW module in parallel

```
module EXAMPLE_SW:
output O;
output P;
input A_and_C_or_B_and_C;
input A_or_B;

present A_and_C_or_B_and_C then
  emit O
end present;
present A_or_B then
  emit P
end present

end module
```

```
module EXAMPLE_HW:
input A, C, B;
output A_and_C_or_B_and_C, A_or_B;

  every immediate [A_or_B and C] do
    emit A_and_C_or_B_and_C
  end every
||
  every immediate [B or A] do
    emit A_or_B
  end every

end module
```

```
module EXAMPLE:
input A, B, C;
output O, P;

signal A_and_C_or_B_and_C,
       A_or_B in
    run EXAMPLE_SW
  ||
    run EXAMPLE_HW
end signal

end module
```

Approach

Second Step: HW/SW Synthesis

- SW module is compiled to KEP assembler code and then executed on the KEP
- HW module is transformed into a VHDL description of a combinational logic
- KEP has to provide an interface to the logic block

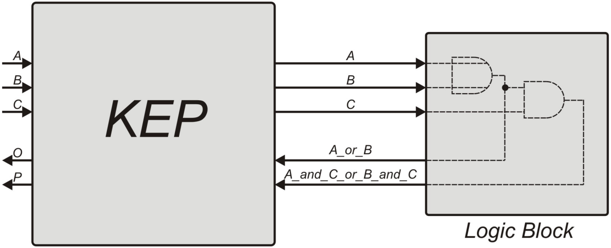

Figure: Schematic of the KEP with external logic block

Experimental Results

Test conditions

- Benchmark: TCINT from the EstBench suite

Results

- KEP code already faster than the fastest MicroBlaze code generated by the Esterel V7 compiler
- Maximum execution time of the co-design has reduced to less than the half of the pure SW solution

		Original program (SW)				Partitioned program (SW)				HW+SW
		MicroBlaze			KEP	MicroBlaze			KEP	KEP
		V5	V7	CEC		V5	V7	CEC		
Memory (in bytes)		14860	**11376**	15340	3527	17308	**11416**	17460	4471	1894
Clock cycles	Average	3488	**1797**	2121	261	4248	**1826**	2971	720	204
	Maximum	3580	**1878**	2350	729	4336	**1907**	3311	981	345
	Empty input	3476	**1807**	2101	237	4267	**1838**	2956	771	204

Table: Experimental results for the TCINT benchmark.