ECPS 203
Embedded Systems Modeling and Design
Lecture 18

Rainer Dömer

doemer@uci.edu

Center for Embedded and Cyber-physical Systems
University of California, Irvine

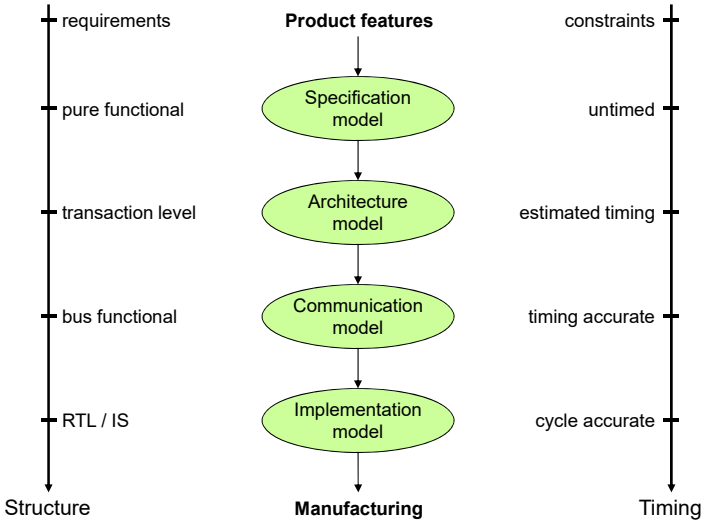
Lecture 18: Overview

- Course Administration
  - Final course evaluation
- Hardware Modeling in Embedded Systems
  - Register Transfer Level (RTL) abstraction
  - RTL synthesis
  - RTL component modeling in SystemC
  - Example modules

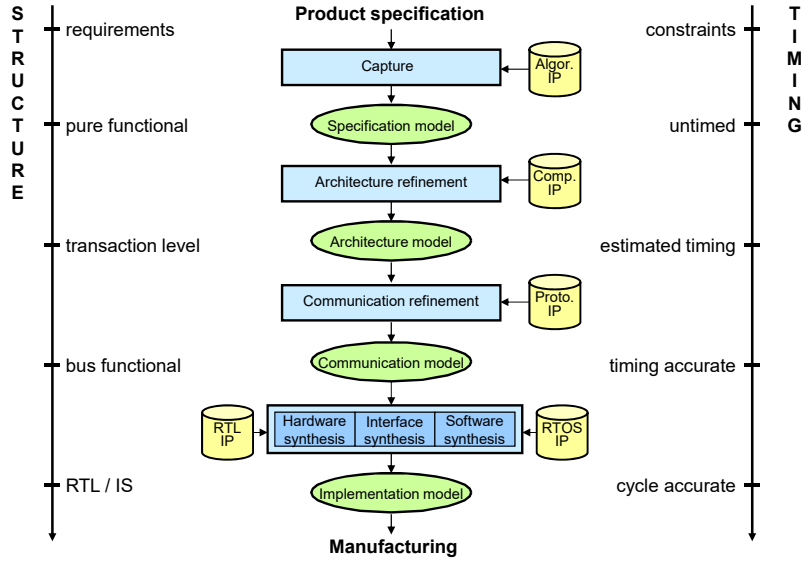
Course Administration

- Final Course Evaluation
  - Open until end of 10th week (Sunday night)
  - Nov. 25, 2019, through Dec. 8, 2019, 11:45pm
  - Online via EEE Evaluation application
- Mandatory Evaluation of Course and Instructor
  - Voluntary
  - Anonymous
  - Very valuable
- Please spend 5 minutes for this survey!
  - Your feedback is appreciated!

Top-Down Design Flow

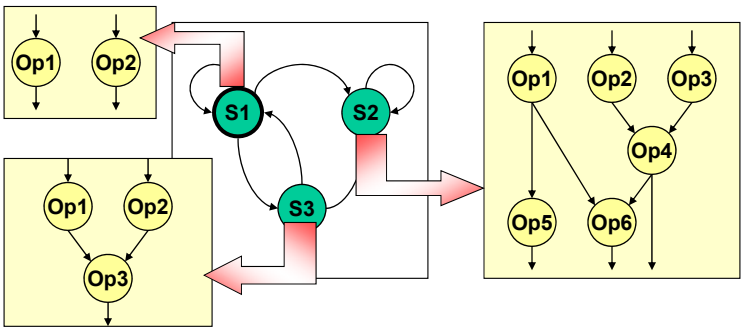


Top-Down Design Flow



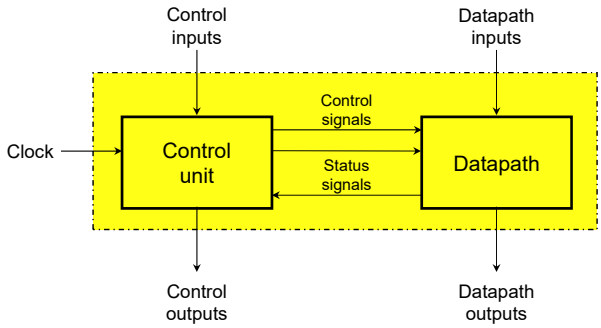
Hardware Modeling in Embedded Systems

- Register Transfer Level (RTL) Abstraction
  - FSMD Model: Finite State Machine with Data
    - Combined model for control and computation
    - Implementation: controller plus datapath



Hardware Modeling in Embedded Systems

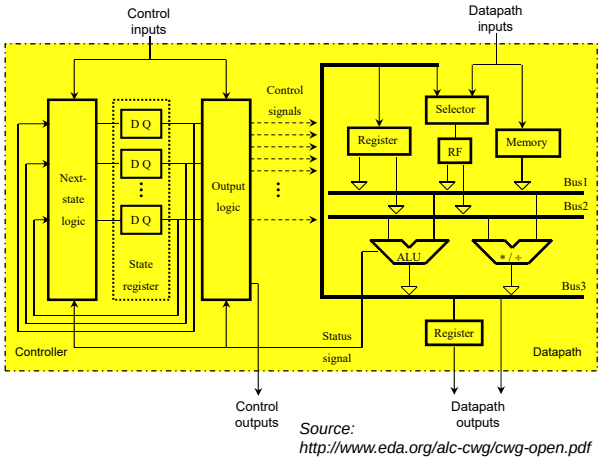
- Register Transfer Level (RTL) Abstraction
  - Block diagram of a generic RTL component (high level)



Source: http://www.eda.org/alc-cwg/cwg-open.pdf

Hardware Modeling in Embedded Systems

- Register Transfer Level (RTL) Abstraction
  - Block diagram of a generic RTL component (low level)



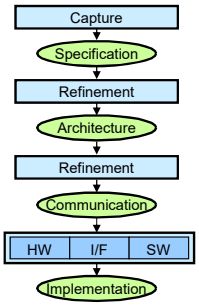
Source: http://www.eda.org/alc-cwg/cwg-open.pdf

Hardware Synthesis in Embedded Systems

- Register Transfer Level (RTL) Synthesis
  - Allocation of RTL components
    - Type and number of functional units
    - Type and number of storage units
    - Type and number of interconnecting busses
  - Scheduling
    - Representation of basic blocks as super-states
    - Scheduling of operations to clock cycles
  - Binding
    - Bind functional operations to functional units
    - Bind variables to storage units
    - Bind assignments/transfers to busses
  - Result:
    - Clock-cycle and pin accurate model at RTL abstraction

Hardware Synthesis in Embedded Systems

- Example: Register Transfer Level Synthesis in SCE
  - Allocation of RTL components
    - Type and number of functional units (e.g. adder, multiplier, ALU)
    - Type and number of storage units (e.g. registers, register files)
    - Type and number of interconnecting busses (drivers, multiplexers)
  - Scheduling
    - Basic blocks assigned to super-states
    - Individual operations assigned to clock cycles
  - Binding
    - Bind functional operations to functional units
    - Bind variables to storage units
    - Bind assignments/transfers to busses
  - ➢ Result
    - Clock-cycle accurate model of each HW PE
    - Output: Synthesizable Verilog description



RTL Modeling in SystemC

- Example Modules of RTL Components
  - D-FF
  - D-FF with Asynchronous Reset
  - Memory
  - Shifter
  - Counter
  - State machine

  - Source: Aleksandar Milenkovic (ECE at UAH)
  - *CPE 626:*
    The SystemC Language – VHDL, Verilog Designer's Guide
  - homepages.cae.wisc.edu/~ece734/SystemC/cpe626-SystemC-L2.ppt

RTL Modeling in SystemC

- Example Modules of RTL Components
  - D-FF

```
// dff.h
#include "systemc.h"
SC_MODULE(dff)
{
  sc_in<bool> din;
  sc_in<bool> clock;
  sc_out<bool> dout;

  void doit()
  {
     dout = din;
  };
 SC_CTOR(dff)
 {
   SC_METHOD(doit);
   sensitive_pos << clock;
 }
};
```

  - Source: Aleksandar Milenkovic (ECE at UAH)

RTL Modeling in SystemC

- Example Modules of RTL Components
  - D-FF with asynchronous reset

```
// dffa.h
#include "systemc.h"
SC_MODULE(dffa)
{ sc_in<bool> clock;
  sc_in<bool> reset;
  sc_in<bool> din;
  sc_out<bool> dout;
  void do_ffa()
  { if (reset) {
       dout = false;
    } else if (clock.event()) {
        dout = din;}
  };
  SC_CTOR(dffa) {
    SC_METHOD(do_ffa);
          sensitive(reset);
          sensitive_pos(clock);
  }
};
```

- Source: Aleksandar Milenkovic (ECE at UAH)

RTL Modeling in SystemC

- Example Modules of RTL Components
  - Memory

```
// ram.h
#include “systemc.h”
SC_MODULE(ram)
{
  sc_in<sc_int<8> > addr;
  sc_in<bool> enable;
  sc_in<bool> readwr;
  sc_inout_rv<16> data;

  void read_data();
  void write_data();
  sc_lv<16> ram_data[256];

SC_CTOR(ram)
{
  SC_METHOD(read_data);
    sensitive << addr << enable << readwr;
  SC_METHOD(write_data);
   sensitive << addr << enable << readwr << data;
}
};
```

- Source: Aleksandar Milenkovic (ECE at UAH)

RTL Modeling in SystemC

- Example Modules of RTL Components
  - Memory (cont'd)

```
// ram.cc
#include "ram.h"
void ram::read_data()
{
  if (enable && ! readwr ) {
    data = ram_data[addr.read()];
  } else {
    data = "ZZZZZZZZZZZZZZZZ";
  }
}

void ram::write_data()
{
  if (enable && readwr) {
    ram_data[addr.read()] = data;
  }
}
```

- Source: Aleksandar Milenkovic (ECE at UAH)

RTL Modeling in SystemC

- Example Modules of RTL Components
  - Shifter

```
// shift.h
#include "systemc.h"
SC_MODULE(shift)
{
  sc_in<sc_bv<8> > din;
  sc_in<bool> clk;
  sc_in<bool> load;
  sc_in<bool> LR;
  sc_out<sc_bv<8> > dout;
  sc_bv<8> shiftval;
  void shifty();
  SC_CTOR(shift)
  {
     SC_METHOD(shifty);
       sensitive_pos (clk);
  }
};
```

```
// shift.cc
#include "shift.h"
void shift::shifty()
{
  if (load) {
     shiftval = din;
  } else if (!LR) {
     shiftval.range(6,0) = shiftval.range(7,1);
     shiftval[7] = '0';
  } else if (LR) {
     shiftval.range(7,1) = shiftval.range(6,0);
     shiftval[0] = '0';
  }
  dout = shiftval;
}
```

- Source: Aleksandar Milenkovic (ECE at UAH)

RTL Modeling in SystemC

- Example Modules of RTL Components
  - Counter

```
#include "systemc.h"
SC_MODULE(counter)
{
  sc_in<bool> clock;
  sc_in<bool> load;
  sc_in<bool> clear;
  sc_in<sc_int<8> > din;
  sc_out<sc_int<8> > dout;
  int countval;
  void onetwothree();

  SC_CTOR(counter)
  {
     SC_METHOD(onetwothree);
          sensitive_pos (clock);
  }
};
```

```
// counter.cc
#include "counter.h"
void counter::onetwothree()
{
  if (clear) {
     countval = 0;
  } else if (load) {
  countval = din.read(); // use read when a
  // type conversion is happening
  // from an input port
  } else {
    countval++;
   }
   dout = countval;
}
```

- Source: Aleksandar Milenkovic (ECE at UAH)

RTL Modeling in SystemC

- Example Modules of RTL Components
  - State Machine
    - Voicemail controller
    - States
      - Main
      - Send
      - Review
      - Repeat, Erase, Record
    - Outputs
      - play, recrd, erase, save and address
    - Two SC_METHOD processes
      - getnextst – calculates the next state based on input and current state
      - setstate – copies the calculated next_state to the current_state every positive clock edge on input clk

- Source: Aleksandar Milenkovic (ECE at UAH)

RTL Modeling in SystemC

- Example Modules of RTL Components
  - State Machine

```
// stmach.h
#include "systemc.h"
enum vm_state {
  main_st, review_st, repeat_st,
  save_st, erase_st, send_st,
  address_st, record_st,
  begin_rec_st, message_st
};
SC_MODULE(stmach)
{
  sc_in<bool> clk;
  sc_in<char> key;
  sc_out<sc_logic> play;
  sc_out<sc_logic> recrd;
  sc_out<sc_logic> erase;
  sc_out<sc_logic> save;
  sc_out<sc_logic> address;
  sc_signal<vm_state> next_state;
  sc_signal<vm_state> current_state;
```

```
  void getnextst();
  void setstate();

SC_CTOR(stmach)
{
  SC_METHOD(getnextst);
     sensitive << key << current_state;
  SC_METHOD(setstate);
    sensitive_pos (clk);
}
};
```

  - Source: Aleksandar Milenkovic (ECE at UAH)

RTL Modeling in SystemC

- Example Modules of RTL Components
  - State Machine (cont'd)

```
// stmach.cc
#include "stmach.h"
void stmach::getnextst()
{
  play = sc_logic_0;
  recrd = sc_logic_0;
  erase = sc_logic_0;
  save = sc_logic_0;
  address = sc_logic_0;
  switch (current_state) {
    case main_st:
      if (key == '1') {
          next_state = review_st;
      } else {
        if (key == '2') {
            next_state = send_st;
        } else {
            next_state = main_st;
        }
     }
```

```
   case review_st:
     if (key == '1') {
         next_state = repeat_st;
     } else {
       if (key == '2') {
          next_state = save_st;
       } else {
         if (key == '3') {
            next_state = erase_st;
         } else {
           if (key == '#') {
             next_state = main_st;
           } else {
             next_state = review_st;
           }
         }
       }
     }
```

  - Source: Aleksandar Milenkovic (ECE at UAH)

RTL Modeling in SystemC

- Example Modules of RTL Components
  - State Machine (cont'd)

```
case repeat_st:
   play = sc_logic_1;
   next_state = review_st;
case save_st:
   save = sc_logic_1;
   next_state = review_st;
case erase_st:
   erase = sc_logic_1;
   next_state = review_st;
case send_st:
   next_state = address_st;
case address_st:
   address = sc_logic_1;
   if (key == '#') {
      next_state = record_st;
   } else {
      next_state = address_st;
   }
```

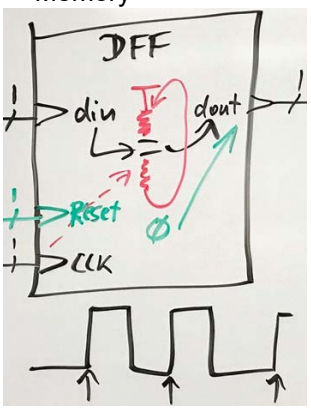
- Source: Aleksandar Milenkovic (ECE at UAH)

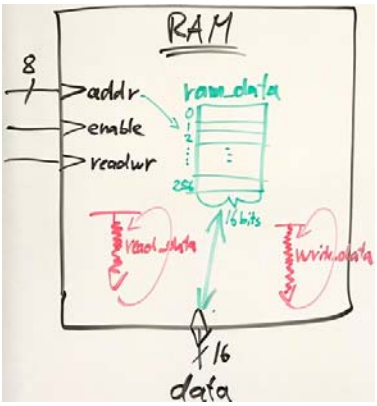
```
   case record_st:
      if (key == '5') {
        next_state = begin_rec_st;
      } else {
        next_state = record_st;
      }
    case begin_rec_st:
      recrd = sc_logic_1;
      next_state = message_st;
    case message_st:
      recrd = sc_logic_1;
      if (key == '#') {
         next_state = send_st;
      } else {
         next_state = message_st;
      }
  } // end switch
} // end method
void stmach::setstate()
{
  current_state = next_state;
}
```

ECPS203: Embedded Systems Modeling and Design, Lecture 18 (c) 2019 R. Doemer 21

RTL Modeling in SystemC

- Example Modules of RTL Components
  - Source: Aleksandar Milenkovic (ECE at UAH)
  - D-FF with/without asynchronous reset
  - Memory

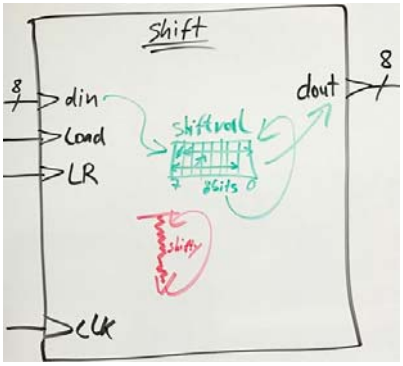


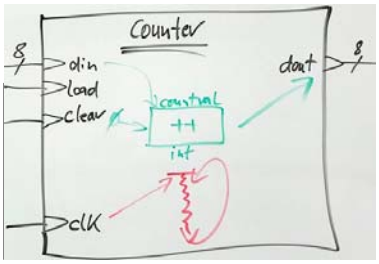


ECPS203: Embedded Systems Modeling and Design, Lecture 18 (c) 2019 R. Doemer 22

RTL Modeling in SystemC

- Example Modules of RTL Components
  - Source: Aleksandar Milenkovic (ECE at UAH)
  - Shifter
  - Counter





RTL Modeling in SystemC

- Example Modules of RTL Components
  - Source: Aleksandar Milenkovic (ECE at UAH)
  - State machine