LUNDS UNIVERSITET
Lunds Tekniska Högskola

Institutionen för datavetenskap
Krzysztof Kuchcinski

Tentamen i kursen
EDAN15: Konstruktion av inbyggda system
(Design of Embedded Systems)

2018-08-21, kl. 8-13

Sal:
Vic: 3C

Hjälpmedel:
Inga

Resultat anslås:
Senast 2018-09-04

Poänggränser:
Max 40 p., för godkännande krävs ca 20 p.

Jourhavande lärare:
Krzysztof Kuchcinski, tel. 22 23414

The answers to the questions can be written in Swedish or English.

Good luck!
Lycka till!

1 (3 p.)

During development of an embedded system a designer needs to consider different kind of implementation trade-offs. Typically one need to consider different implementation options and their effect on the final design. Discuss three typical design trade-offs between design parameters. Point out what is the effect on the final design.

2 (3 p.)

Draw the data-flow graph for the following code assuming that each actor may contain at most one arithmetic operation.

```
x <= a + b;
case x is
  when 0 => y <= a + 1;
  when 1 => y <= a - 1;
  when 2 => y <= 2 * a;
  when others => y <= b;
end case;
```

In your model, you have to use the following actors.

- actors implementing addition of two numbers ($a + b$) and shift left operation ($a << n$), where n is a constant,
- equal zero ($= 0$); producing output *true* or $false$ depending on the input value being $= 0$ or $\neq 0$ respectively,
- greater then ($> n$); producing output *true* or $false$ depending on the input value being $> n$ or $\leq n$ respectively,
- selector; selecting $input_1$ or $input_2$ depending on *sel* input being *true* or $false$.

3 (3 p.)

Petri net is a formalism that can be used to create abstract models of embedded systems. Explain basic primitives used in Petri nets. What are execution rules for marked Petri nets? In your discussion explain also how firing of a transition is defined and what does it mean that a transition is *enabled* and it *may* fire?

Define Petri nets for a) sequential execution, b) parallel execution and c) non-deterministic choice.

4 (5 p.)

Translate the C function `foo(int, int)`, shown in listing 1 to the VHDL entity shown in listing 2. Use `std_logic_vector(0 to 31)` for the integers. The VHDL implementation should be a mapped to *combinatorial* logic when synthesized. The implementation are to be done using behavioral synthesizable VHDL.

Listing 1: Translate the following C code to VHDL

```
int foo(a, b){
```

```
 tmp = a+b;
 if(tmp > 30)
   return tmp - 30;
 } else {
   if(a>b){
     return a;
   } else {
     return b;
   }
 }
}
```

Listing 2: Write the behavior of the FOO entity

```
entity FOO is
port (
  a, b: in STD_LOGIC_VECTOR(0 to 31);
  result : out STD_LOGIC_VECTOR(0 to 31);
);
end FOO;
```

5 (6 p.)

Implement a VHDL component that finds the position of every occurrence of the word "SOS" in a sequence of characters. The characters are inputed on the input `token`, one character each clock cycle. If the character on `token` is the second S in "SOS", then the output `match` should be 1, else it is 0. The output `position` should be the position of the beginning of the last found SOS in the character sequence, 1 being the first character, 2 the second etc. 'S' has the token value 83 and 'O' is 79. The implementation are to be done using behavioral synthesizable VHDL.

For example, for the input "SOSOS, SOS was replaced by MAYDAY in the 1920s." `position` should have the values: **0** (clk 1-2), **1** (clk 3-4), **3** (clk 5-10) **8** (clk 10-...), `match` should be **1** at cycle 3, 5, and 10, **0** at all other times.

Listing 3: Write the behavior of the SOS entity

```
entity SOS is
port (
  clk : in STD_LOGIC;
  reset : in STD_LOGIC; -- active low
  token: in STD_LOGIC_VECTOR(0 to 7);
  match : out STD_LOGIC;
  position : out STD_LOGIC_VECTOR(0 to 31);
);
end SOS;
```

6 (4 p.)

System partitioning can be done using partitioning or clustering algorithms. These algorithms use either *objective* or *closeness* functions to make appropriate decisions on system partitioning. Answer the following questions, assuming that our goal is to minimize the communication cost between partitions or clusters.

a) How are *objective* or *closeness* functions defined?

b) How partitioning or clustering objectives, such as minimum communication between partitions, are reflected in objective or closeness functions? Give examples for both cases.

Give examples of objective and closeness function in your discussion.

7 (5 p.)

Using list scheduling, make the schedule for the data dependency graph depicted in Figure 1. Assume that you can use two adders and two pipelined multipliers. Adders have 1 clock cycle delay and multipliers 2 clock cycles delay (two pipeline stages, 1 cycle for each pipeline stage). Answer the following questions:

a) How do you compute the priorities? Write down the priority for each operation.

b) What is the number of clock cycles for execution of this model?

Give the sequence of steps taken by the list scheduling algorithm that lead to your solution. For each step specify the list of nodes that were considered for scheduling.

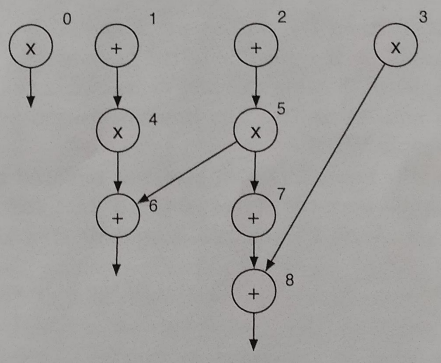

Figure 1: An example of data dependency graph

8 (5 p.)

Describe briefly Earliest Deadline First scheduling (EDF). The presentations should include the following parts:

a) describe the task model and execution assumptions,
b) the method to assign priorities to tasks, and
c) discussion whether EDF can provide the optimal schedule and the condition for scheduliability of tasks.

Draw the EDF schedule for the task set from Table 1 for the first 12 times units.

Task	Period	WCET
1	3	1
2	6	2
3	12	3

Table 1: Task set for EDF scheduling.

9 (3 p.)

What is the formula for power consumption in CMOS technology? Discuss how the power consumption of a design can be minimized considering the parameters of this formula. Specifically, explain how parallelization of computations can be used, not only to speed-up a design, but also to reduce its power consumption.

10 (3 p.)

Discuss briefly the main idea of BIST testability improvement method. In your discussion explain the following parts of BIST:

a) pattern generator,
b) signature analyzer, and
c) BIST controller.

What is the testing procedure and how do you decide whether the unit under test is correct or not. What are advantages and drawbacks of this method?