



LUNDS UNIVERSITET

Lunds Tekniska Högskola

Institutionen för datavetenskap

Krzysztof Kuchcinski

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

2017-06-01, kl. 8-13

Sal:

Sparta A-C

Hjälpmedel:

Inga

Resultat anslås:

Senast 2017-06-20

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.)

Implementation of embedded systems using single or multi-processor systems is very appealing and interesting design choice since software is easier to develop, maintain and upgrade than hardware. This is, however, not always possible. Often a special hardware needs to be developed and included in the final product. Give the motivation why the specialized hardware needs to be included. Enumerate at least two main reasons for this and discuss them briefly.

2 (3 p.)

Explain the notion of *actors*, *tokens* and *firing rules* for data-flow networks built of actors. Describe informally the execution rules for actors.

Use the following two actors to illustrate these notions.

- an actor with two input ports that implements addition of three numbers. Two consecutive numbers coming from the first port and the third number comes from the second port.
- a non-deterministic merger of two input streams.

Define the firings rules for these actors.

3 (3 p.)

Model, using Petri nets, an asynchronous processor control unit with the following characteristics.

- The processor has two registers R0, R1 and two functional units A and B,
- Ready to execute instructions are in buffer I,
- Instructions are issued to be executed when the registers and the functional units they need are available (not used by other instructions),
- there are two types of instructions
 - instruction type 1 requires register R1 and functional unit A to execute, and
 - instruction type 2 requires registers R0, R1 and functional unit B to execute.

For drawing your solution start from the partial Petri net structure depicted in Figure 1. Add back edges, if needed.

4 (5 p.)

Write synthesizable behavioral VHDL code (both entity and architecture) for a component with the inputs and outputs as depicted in Figure 2. Use IEEE.STD_LOGIC_1164.

The component should have the following behaviour.

sel	q	comment
00	a + b	addition
01	a sll b	shift left logical
10	a or b	logical or
11	a and b	logical and

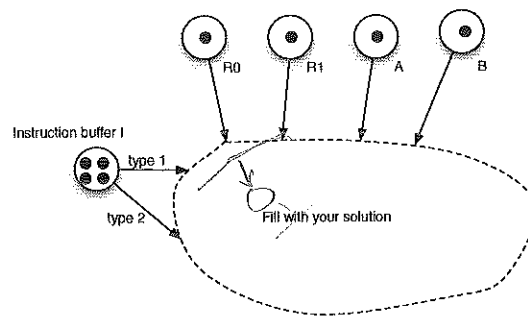


Figure 1: Partial Petri net structure for assignment 3.

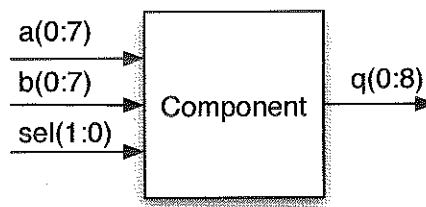
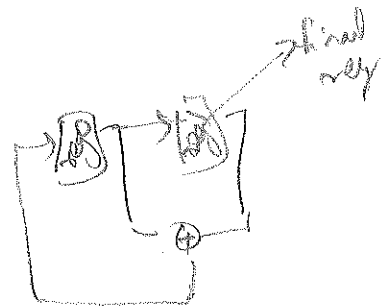


Figure 2: The component.



5 (6 p.)

Implement a VHDL component that computes the n :th Lucas number. The Lucas sequence is defined as $L_0 = 2, L_1 = 1, L_n = L_{n-1} + L_{n-2}$ for $n \geq 2$ (the sequence starts with: 2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, ...). The entity of the component is shown in Listing 1. A new computation must start when `start` is high at a *rising clock edge* (`start` will only be high for one clock cycle). The computation is expected to take several clock cycles (about $n - 2$ clock cycles). If the Lucas component is busy with another computation at the time `start` goes high, the old computation is to be aborted. When the computation is finished, `done` should go high and `lucas` should contain the n :th Lucas number. You may keep `done` high any number of clock cycles (at least one) when the computation is finished. During the computation `done` must be low. You may assume $n \geq 2$ and not changing during the computation. You may also ignore overflow, i.e. the answer will fit in 32 bits.

clk'event & clk = '1' then
 if (start = '1') then
 reg0 <= 2;
 reg1 <= 1;
 done <= '0';
 else
 reg0 <= reg1;
 reg1 <= reg0 + reg1;
 end if;
end if;

Listing 1: Write the behavior for the lucas entity

```

entity LUCAS is
port (
  clk : in STD_LOGIC;
  n: in STD_LOGIC_VECTOR(0 to 31);
  start : in STD_LOGIC;
  lucas : out STD_LOGIC_VECTOR(0 to 31);
  done : out STD_LOGIC;
);
end LUCAS;

```

6 (4 p.)

Assume that the application graph, specified as a weighted graph, depicted in Figure 3 represents the tasks and their intercommunications for a certain application. The weight of an edge represents the communication "cost" while the weight of a node is the "size" of the task in a partition. Assuming that we want to use a clustering algorithm which will group together tasks, give an expression which defines a closeness function for such an algorithm. The closeness function has to make it possible to minimize the communication cost between partitions.

Give the value of your closeness function between the following two clusters for these three cases.

1. {3} and {5},
2. {1} and {2},
3. {6, 7} and {4, 8, 9}.

Which of these three cases is the best to select for clustering in the next step of clustering algorithm according to your closeness function?

Make the hierarchical clustering of this graph using your closeness function.

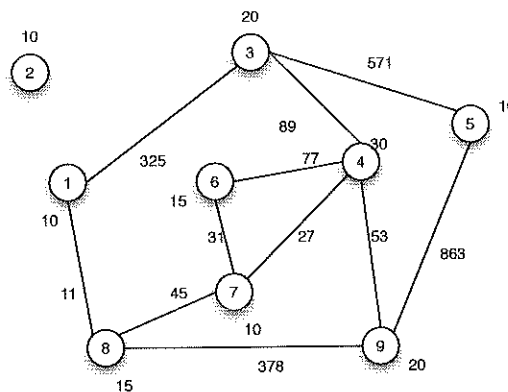


Figure 3: An example of an application graph for assignment 6.

7 (5 p.)

Using list scheduling, make the schedule for the data dependency graph depicted in Figure 4. Assume that you can use one adder 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:

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

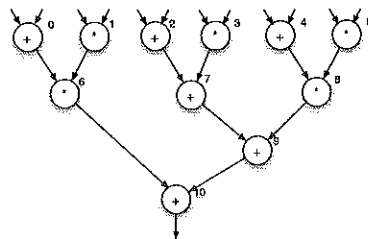


Figure 4: An example of data dependency graph

8 (5 p.)

Describe briefly Rate Monotonic Scheduling (RMS). The presentations should include the following parts:

- describe the task model and execution assumptions,
- the method to assign priorities to tasks, and
- discussion whether RMS can provide the optimal schedule and the condition for schedulability of tasks.

Draw the RMS 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	2

Table 1: Task set for RMS 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?