LUNDS UNIVERSITET
Lunds Tekniska Högskola

Institutionen för datavetenskap
Krzysztof Kuchcinski

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

2016-06-02, kl. 14-19

Sal:
Sparta:A,B

Hjälpmedel:
Inga

Resultat anslås:
Senast 2016-06-16

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.)

Explain the term "design space exploration" used in embedded systems design. What does it mean that designers make "trade-offs" between different parameters? Give three parameters usually considered. Illustrate this with examples.

2 (3 p.)

The handshaking protocol specifies how data are communicated between Master and Slave units. The protocol has the following phases.

1. Master asserts signal `req` to receive data,
2. Slave puts data on bus and asserts signal `ack`,
3. Master receives data and deasserts signal `req`, and
4. Slave deasserts `ack` and is ready for next request.

Formalize this specifications using Petri nets and prove that the master after sending request (`req`) will get data. Draw the example for two slaves.

3 (3 p.)

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

$$y_n = y_{n-1} + (x_n + y_n) * z_n$$

Specify the firing rules and functions for each actor in your model. A function specifies an actor output as a function of its inputs, i.e., $out = f(inputs)$.

4 (6 p.)

Implement a VHDL component that computes the *n*:th Horadam number. The Horadam sequence is defined by four constants (p, q, r, s), the initial values $H_0 = p$ and $H_1 = q$ and the equation

$$H_n = s \cdot H_{n-1} + r \cdot H_{n-2}$$

where $p = 0$, $q = 1$, $r = 2$ and $s = 3$, for example. These values must be specified in the VHDL code as parameters. The sequence starts with: 0, 1, 2, 5, 12, 29, 70, 169, 408, 985, 2378, 5741, ...

The interface of the component is shown in the entity at 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 Horadam component is busy with another computation at the time `start` goes high, the old computation must be aborted. When the computation is finished, `done` should go high and `horadam` should contain the *n*:th Horadam 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.

Listing 1: Write the behavior for the horadam entity

```
entity HORADAM is
port (
  clk : in STD_LOGIC;
  n: in STD_LOGIC_VECTOR(0 to 31);
  start : in STD_LOGIC;
  horadam : out STD_LOGIC_VECTOR(0 to 31);
  done : out STD_LOGIC;
);
end HORADAM;
```

5 (5 p.)

You are to implement a 2 tap IIR filter in VHDL. The function of the IIR filter is $Y_n = X_n + \frac{1}{2}Y_{n-1} + \frac{1}{4}Y_{n-2}$. X is the input, and Y is the output. The VHDL entity for the component is shown in listing 2. You do not need to handle overflow, i.e. you can assume that the output will always fits in 32 bits. *reset* is active high. At a reset the output and the internal state must be set to 0 (Y_n, Y_{n-1}, and Y_{n-2} in the formula). Your implementation must be written in synthesizable behavioral VHDL code.

Listing 2: Write the behavior of for the IIR entity

```
entity IIR is
port (
 clk : in STD_LOGIC;
 reset : in STD_LOGIC;
 X : in STD_LOGIC_VECTOR(0 to 31);
 Y : out STD_LOGIC_VECTOR(0 to 31);
);
end IIR;
```

6 (4 p.)

Assume that the weighted graph, depicted in Figure 1, represents tasks and their intercommunications. The weight of an edge (w_{ij}) represents the communication "cost" between nodes i and j, while the weight of a node (v_i) is the "size" of task i. Using this example graph

- define a *cost function* for a partitioning algorithm for minimization of communication cost, and
- define conditions that limit the difference between the sizes of two partitions to maximally 5%.

Give the value of your cost function for two different partitionings

1. partition 1: P1, P2, P3, P6, P8 and partition 2: P4, P5, P7, P9
2. partition 1: P1, P2, P3, P7 and partition 2: P4, P5, P6, P8, P9

Which partitioning of the two specified above is better?

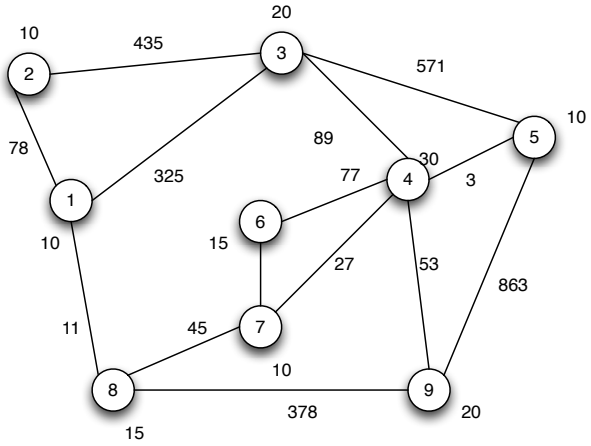

Figure 1: An example system represented as a graph.

7 (5 p.)

Using list scheduling, make the schedule for the data dependency graph depicted in Figure 2. 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:

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 that lead to your solution. For each step specify the list of nodes that were considered for scheduling.

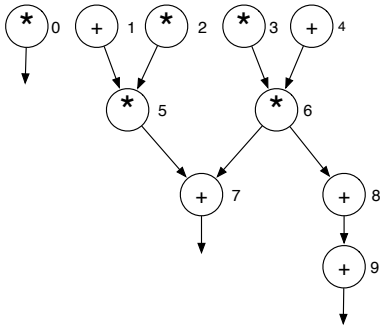

Figure 2: An example of data dependency graph

8 (5 p.)

Describe briefly Rate Monotonic Scheduling (RMS). 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 RMS can provide optimal schedule and the condition for schedulability of tasks.

Draw a RMS schedule for the task set from Table 1.

Task	Period	WCET
1	6	1
2	12	3
3	24	5

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 SCAN path testability improvement technique. In the discussion include the following points:

a) the general idea of the SCAN path, and
b) give the motivation why SCAN path improves test generation and testing time.