Seminars in Embedded System Design
EDAN15

Krzysztof Kuchcinski
Department of Computer Science
Lund Univeristy

April 11, 2011

1 Petri net

Draw simple Petri nets which model sequential execution, parallel execution and non-deterministic choice.

2 Petri net

Figure 1 models the situation when two tasks want to get exclusive access to a shared resource, in our example a printer. Model, using Petri nets, the part of the figure which is depicted by a “cloud”. This part, in response to request signals from tasks (request1 or request2) has to grant access to the printer to a single task and generate an acknowledge signal (acknowledge1 or acknowledge2). Each task releases the shared resource by generating release signals (release1 or release2).

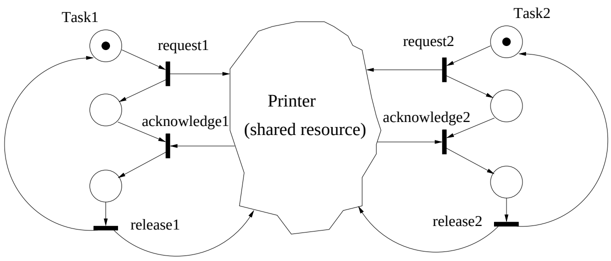

Figure 1: A model of a two tasks system with a shared resource.

3 Data-flow model

Using data-flow networks model a simple filter which reads in each clock cycle one unsigned integer value form input in and outputs a value equal an average over the last four read values. This means $(in_{-3} + in_{-2} + in_{-1} + in_0)/4$. Explain what actors you use and what are their firing rules.

4 Data-flow model

Model the following conditional computation of the form
`if c then y = x + 1 else y = x - 1`
using Boolean data-flow (BDF) actors. You will probably need to use the typical data-flow actors such as “+” and “-” as well as specific BDF actors such as “select”.

5 Data-flow model

What is a firing rule for a data-flow network? Give the firing rules for the following actors:

a) an adder which adds two operands,
b) a selector which selects either *input 0* or *input 1* and sends it to the *output* depending on the value on the input *select*, and
c) a merger which merges either three or two inputs from the set of inputs *input 0*, *input 1*, and *input 2* into a single *output*. The decision to select two inputs *input 0* and *input 1* or all three inputs is made based on the value of input *number* which specifies either 2 or 3. If input *number* has a different value the actor does not do anything.

6 VHDL

Give the values assigned to signals and variables by the part of the VHDL process included below.

```
P1: process
    variable a, b : integer;
begin
       :
   -- initial values of s1 = 1, s2 = 0, a = 0, b = 0
   s1 <= 10;
   a := s1;
   b := 11;
   s2 <= b + s1;
   -- give values of s1, s2, a, b here (1)
   wait for 10 ns;
   -- give values of s1, s2, a, b here (2)
       :
end process;
```

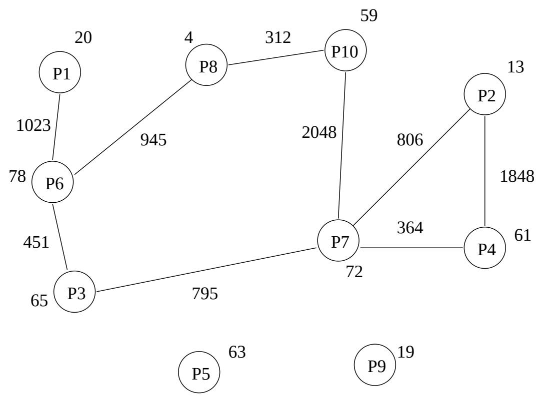

Figure 2: An example system represented as a graph.

7 Clustering

Assume that the weighted graph, such as depicted in Figure 2, represents tasks and their intercommunications. 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 into clusters, give an expression which defines a closeness function for such an algorithm. The closeness function has to make it possible to minimize the communications cost between partitions.

Give the value of your closeness function for the following sets of two nodes

1. P2 and P3,
2. P8 and P10,
3. P3 and P7.

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

8 Partitioning

Assume that the weighted graph, such as depicted in Figure 2, represents tasks and their intercommunications. 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 partition the task graph into two partitions, give an expression which defines a cost function for a partitioning algorithm. The cost function has to make it possible to minimize the communications cost while making partitions of a similar size (as much as possible). How can you control the minimization of communication costs vs. keeping the sizes of partitions similar?

Give the value of your cost function for two different partitionings

1. partition 1: P2, P3, P4, P7, P10 and partition 2: P1, P5, P6, P8, P9
2. partition 1: P2, P4, P7, P9, P10 and partition 2: P1, P3, P5, P6, P8

Which partition of the two specified above is better?

9 List scheduling

Using list scheduling, make a schedule for a data-flow graph depicted in Figure 3. Assume that you can use one adder and two multipliers. Adders have 1 clock cycle delay and multipliers 2 clock cycles delay. Answer the following questions:

a) What priorities do you use in your list scheduling?
b) What is the number of clock cycles for execution of this model?

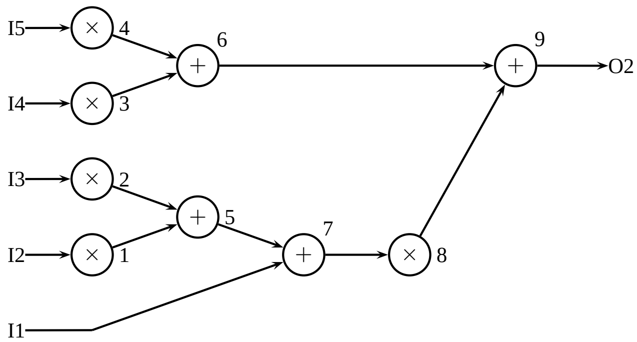

Figure 3: An example of data-flow graph

10 Scheduling

Make a static resource-constraint schedule of the example depicted in Figure 4. Available resources are one adder with 1 clock cycle delay and one multiplier with 2 clock cycles delay. What is the number of clock cycles for a single execution of the graph?

11 Force-directed scheduling

Using forced-directed scheduling, make a schedule for a data-flow graph depicted in Figure 4. Assume that adders have 1 clock cycle delay and multipliers 2 clock cycles delay. What is the minimum number of adders and multipliers for a single execution of the graph in 5 clock cycles?

12 RMS analysis

Use the RMS analysis to check sufficient condition for the following task sets schedulability with RMS. Are the tasks schedulable?

0.2 + 0.267 + 0.2 = 0.667 < 0.779 OK
1/3 + 1/5 + 1/6 + 2/10 = 0.889 > 0.756 NOT but schedulable !?

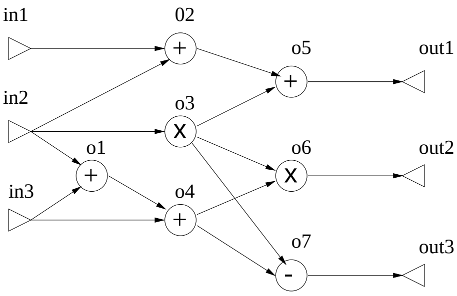

Figure 4: An example of data-flow graph

Task set 1

Task	Period	Worst execution time
1	100	20
2	150	40
3	300	60

Task set 2

Task	Period	Worst execution time
1	3	1
2	5	1
3	6	1
4	10	2