Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2019-06-13
Sal (1)	TER2(2)
Tid	8-12
Utb. kod	TDDI08
Modul	TEN1
Utb. kodnamn/benämning **Modulnamn/benämning**	Konstruktion av inbyggda system En skriftlig tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	12
Jour/Kursansvarig Ange vem som besöker salen	Petru Eles
Telefon under skrivtiden	013-281396
Besöker salen ca klockan	9.30
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Veronica Kindeland Gunnarsson 013-285634 veronica.kindeland.gunnarsson2liu.se
Tillåtna hjälpmedel	English dictionary
Övrigt	
Antal exemplar i påsen	

LINKÖPINGS TEKNISKA HÖGSKOLA
Institutionen för datavetenskap
Petru Eles

Tentamen i kursen

Embedded Systems Design - TDDI08

2019-06-13, kl. 8-12

Hjälpmedel:

Engelsk ordbok.

Supporting material:

English dictionary.

Poänggränser:

Maximal poäng är 30.
För godkänt krävs sammanlagt 16 poäng.

Points:

Maximum points: 30.
In order to pass the exam you need a total of minimum 16 points.

Jourhavande lärare:

Petru Eles, tel. 013-28 13 96

Good luck !!!

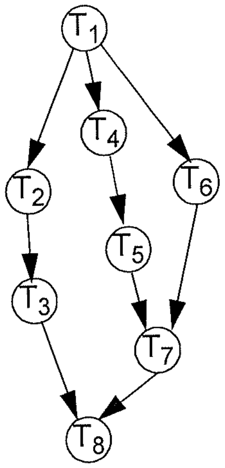
Tentamen i kursen Embedded Systems Design- TDDI08, 2019-06-13, kl. 8-12
Du kan skriva på svenska eller engelska!

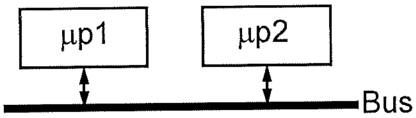
1. Consider an application modelled as the task graph below. Each task, when activated, consumes one message on each input edge and generates, at termination, one message on each output edge. The task graph is executed on the architecture shown in the figure. Execution times of the tasks, when executed on the corresponding processor, are shown in the table. All messages transmitted over the bus, between tasks mapped on different processors, consume 2 time units to reach the destination. Communication between tasks mapped to the same processor is considered to not consume any time.

   Propose an efficient task mapping (indicate on which processor each task is executed) and a corresponding static (nonpreemptive) schedule for the application. Illustrate your schedule as a Gantt chart (similar to the way we captured schedules in Lecture 1&2).

   Try to achieve a maximum delay (the time interval between the start of T1 and the finishing of T8) of 46 time units!

(3p)





Task	WCET	
	μp1	μp2
T_1	5	6
T_2	8	10
T_3	4	5
T_4	5	6
T_5	3	4
T_6	10	11
T_7	17	21
T_8	10	14

2. a) Formulate the synchrony hypothesis for FSMs. What does it imply?

   b) Under which assumptions can we correctly implement a synchronous FSM model?

(2p)

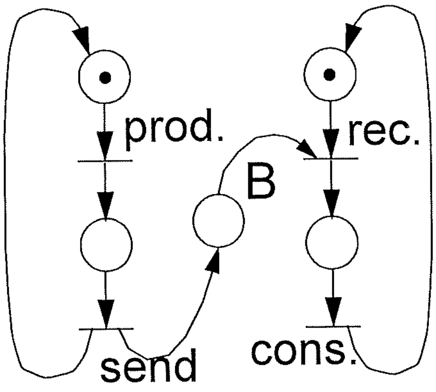
3. Define Kahn process networks and synchronous dataflow models.

   Give an example of a Kahn process network. Show that it cannot be statically scheduled.

   Adjust the example such that it becomes a synchronous dataflow model. Show a static schedule for this new model.

(3p)

4. The figure below represents a Petri Net model for two processes, a producer and a consumer, which are communicating through a buffer; the buffer is represented by place B.



a) Is this Petri Net model bounded?

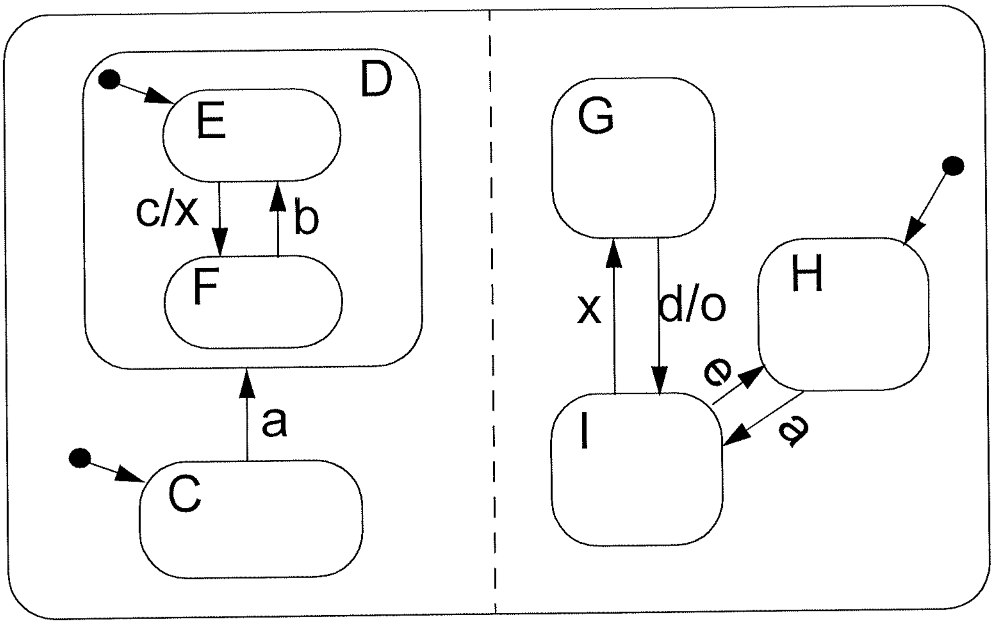
b) How large is the buffer?

c) Which transitions are enabled in this state of the model and why?

d) Draw a similar model in which the buffer has a dimension of four slots.

(3p)

5. Consider the hierarchical concurrent FSM below. Initial states are indicated by the following sign:●⟶. Hence, initially the FSM is in the state C and H. Assume the following sequence of events: a, c, d, b. Indicate the state of the model after each of the four events.



(3)

Tentamen i kursen Embedded Systems Design- TDDI08, 2019-06-13, kl. 8-12
Du kan skriva på svenska eller engelska!

6. How does a discrete event simulator work?
Illustrate by a flow-graph.

(2p)

7. What does it mean by an Application Specific Instruction Set Processor (ASIP)?
We have discussed five dimensions of specialization for ASIPs. Which are those five?
Comment on each of them.

(3p)

8. Describe a simple design flow for processor specialization. Illustrate also by a figure. Comment on the design tools you need.

(2p)

9. We have introduced Systems on Chip with a dynamically reconfigurable datapath; this datapth can be reconfigured to act as an accelerator for the actual program running on the processor. What are the main steps for compiling the source code for such a system? What will result as the outcome of this compilation?

(2p)

10. What does it mean by IP (core) based design? What types of cores can you choose from? Comment on each of them.

(2p)

11. a) Formulate the scheduling problem for a set of real-time tasks.
b) What does it mean that a task set is schedulable?
c) What does it mean by preemptive and non-preemptive scheduling?

(2p)

12. a) What is the basic principle for task scheduling on DVS processors?
b) What is the problem if we consider particularities, concerning power consumption, of individual tasks?
c) How do we solve the problem that only discrete voltage levels are available?
d) Discuss what the problems are if leakage energy is ignored.

(3p)