

ASIC

TENTAMEN TSTE81

Tid: Fredag 12 mars 2004 kl. 14:00–18:00

Plats: GARN

Ansvarig lärare: Oscar Gustafsson, 013-28 16 76, 0704-47 26 17

Hjälpmedel: Räknedosa, Tabell- och formelsamling i aktiva och tidsdiskreta filter, Formelsamling i Kretsteori samt allmänna tabellverk.

Anvisningar: För godkänd tentamen fordras ca 30 poäng.

Total points on first question = $\min \{ 10, (\text{oral exam points} + \text{first question points}) \}$

That is, you are free to solve an arbitrary number of the subproblems of question 1. Total points on the first question will never exceed 10.

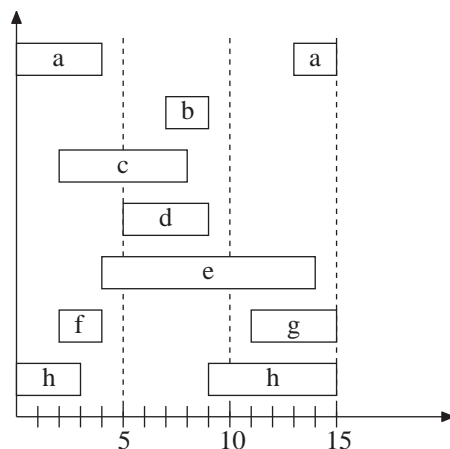
Visning: 29 mars 2004 kl. 12:30-13 i Oscar Gustafssons tjänsterum.

Lösningar: Anslås på ISYs kurstavla i korridor C mellan B25 och B27.

Betygslista: Anslås senast 19 mars 2004 p s s som ovan.

1.
 - a) Vad är vinsten med att använda Booth-kodning för en bit-parallell multiplikator?
What is gain of using Booth coding for a bit-parallel multiplier? (2)
 - b) Ange två redundanta talrepresentationer.
Name two redundant number systems. (2)
 - c) Är FFT en rekursiv algoritm? Motivera!
Is the FFT a recursive algorithm? Motivate your answer. (2)
 - d) Ange två negativa effekter av datarepresentation med ändlig ordlängd.
Name two negative effects of data representation with finite word-length. (2)
 - e) Ange två metoder för att minska problemet med minnesbandbredd i arkitekturer med delat minne.
Name two methods that reduce the memory bandwidth bottleneck for shared memory architectures. (2)

2. Livstidsdiagrammet för ett antal variabler visas nedan.
The lifetime graph for a number of variables is shown below.

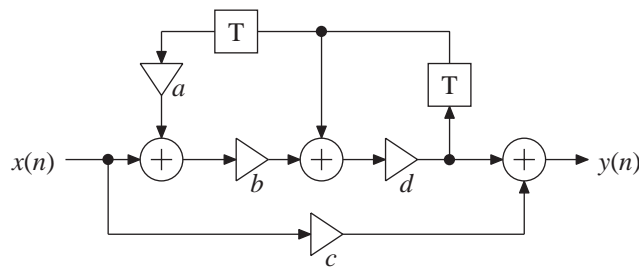


Utför resursallokering och tilldelning för variablerna. Använd left-edge algoritmen och redovisa alla mellansteg. Antag att skrivning och läsning **ej** kan ske i samma tidslucka för en minnescell, dvs minnesvariablerna får inte ligga “kant-i-kant”.

*Perform resource allocation and assignment for the variables. Use the left-edge algorithm and show all intermediate steps. Assume that it is **not** possible to read from and write to a memory cell in the same time slot, i.e., memory variables can not be placed “edge-to-edge”.* (6)

3. Filtret nedan skall implementeras med en arkitektur som använder delat minne. Beräkningselementen är icke-homogena och non-preemptive. Latencyn är två tidsenheter för multiplikationerna a och b , fyra tidsenheter för multiplikationerna c och d , och en tidsenhet för additioner.

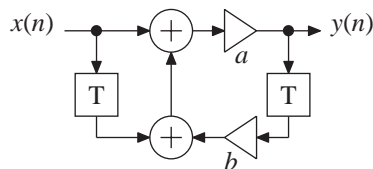
The filter below is to be implemented using a shared memory architecture. The processing elements are non-homogenous and non-preemptive. The latency is two time units for the multiplications a and b , four time units for the multiplications c and d , and one time unit for the additions.



- Beräkna minimala samplingsperioden T_{min} .
Determine the minimal sample period T_{min} . (4)
- Beräkna tiden för den kritiska vägen T_{cp} .
Determine the time of the critical path T_{cp} . (2)
- Rita signalflödesgraf i precedensform.
Draw the signal-flow graph in precedence form. (6)
- Schedulera algoritmen så att $T_{sample} = T_{min}$.
Schedule the algorithm so that $T_{sample} = T_{min}$. (8)

4. Använd aritmetiska transformationer för att minska den kritiska loopen i nedanstående filter till en addition och en multiplikation. Det totala antalet additioner och multiplikationer får inte ökas.

Apply arithmetic transformations to reduce the critical loop in the filter below to one addition and one multiplication. The total number of additions and multiplications should not be increased. (6)



5. Filtret i uppgift 3 skall implementeras med distribuerad aritmetik. Koefficienterna har följande värden: $a = -0.25$, $b = 0.375$, $c = 0.875$ och $d = 0.375$. Minneskodning behöver **inte** användas för att minska storleken hos ROM.

*The filter in problem 3 is to be implemented using distributed arithmetic. The coefficients have the following values: $a = -0.25$, $b = 0.375$, $c = 0.875$, and $d = 0.375$. Memory coding to reduce the size of the ROM is **not** required.*

- a) Beskriv filtret på numeriskt ekvivalent tillståndsform.
Describe the filter using a numerically equivalent state-space representation. (2)
- b) Beskriv arkitekturen. Använd byggblock som skiftackumulatorer, skiftregister, ROM, etc.
Describe the architecture. Use building blocks such as shift accumulators, shift registers, ROM, etc. (6)
- c) Bestäm innehållet i ROM:et/en. Beskriv med lämplig binär representation.
Determine the contents of the ROM(s). Describe the contents using a suitable binary representation. (6)
- d) Vilken latency har de distribuerade aritmetikenheterna?
What is the latency of the distributed arithmetic units? (2)
- e) Bestäm minimal sampelperiod, T_{min} , i klockcykler för filtret baserat på distribuerad aritmetik.
Determine the minimal sample period, T_{min} , in clock cycles for the filter based on distributed arithmetic. (4)
- f) Bestäm minimal sampelperiod, T_{min} , i klockcykler om filtret i stället implementeras direkt med separata bit-seriella multiplikatorer och adderare. Antag modell 0, dvs ingen pipelining efter operationerna.
Determine the minimal sample period, T_{min} , in clock cycles if the filter is implemented directly using separate bit-serial multipliers and adders. Assume Model 0, i.e., no pipelining after the operations. (4)
- g) Rita ett schema för de två multiplikationerna b och c genom att använda en förenklad bitseriell/parallel multiplikator baserad på CSD representation. Använd byggblock som fulladderare, D-vippor och logiska grindar.
Draw the schematic for the two multiplications b and c by using a simplified bit-serial/parallel multiplier based on CSD representation. Use building blocks such as fulladders, flip-flops, and logic gates. (4)