

ASIC

TENTAMEN TSTE81

Tid: Fredag 11 mars 2005 kl. 13:00–17:00

Plats: Nollstället

Ansvarig lärare: Oscar Gustafsson, 013-28 40 59, 0704-47 26 17

Hjälpmedel: Räknedosa.

Anvisningar: För godkänd tentamen fordras 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.

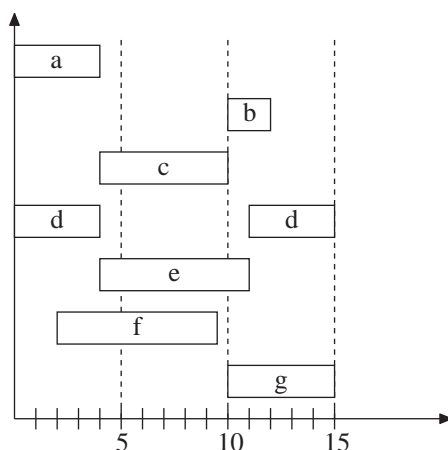
Visning: 17 mars 2005 kl. 12:30-13 i Oscar Gustafssons tjänsterum.

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

Betygslista: Anslås senast 17 mars 2005 p s s som ovan.

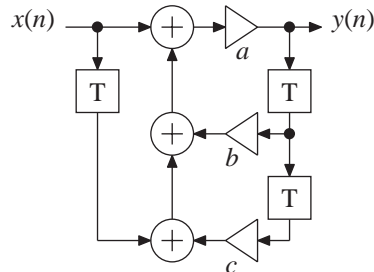
1.
 - a) Ange och beskriv två typer av optimalitet för schemuleringar.
Name and describe two types of optimality for scheduling. (2)
 - b) Vad är vinsten med att använda Booth-kodning för en shiftackumulatorbaserad multiplikator?
What is the gain of using Booth coding for a shift-accumulator based multiplier? (2)
 - c) Vad skiljer en iterativ och en blockbehandlande DSP-algoritm?
What is the difference between an iterative and a block processing DSP algorithm? (2)
 - d) Vilka variabler finns alltid i den första nodmängden i en precedensgraf?
Which variables are always put in the first node set in a precedence graph? (2)
 - e) Hur beror antal operationer i en FFT på antalet punkter som beräknas?
How does the arithmetic workload depend on the length of the sequence for an FFT? (2)

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

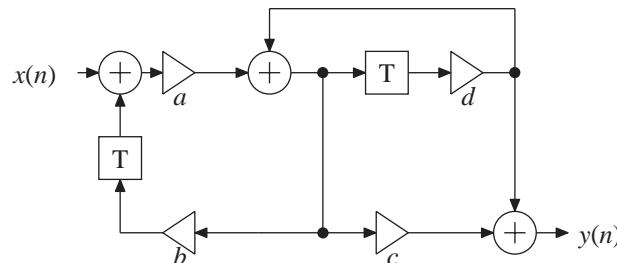


Utför resursallokering och tilldelning för variablerna med klickpartitionering (clique partitioning). Antag att skrivning och läsning kan ske i samma tidslucka för en minnescell, dvs minnesvariablerna får ligga “kant-i-kant”.
Perform resource allocation and assignment for the variables using clique partitioning. Assume that it is possible to read from and write to a memory cell in the same time slot, i.e., memory variables can be placed “edge-to-edge”. (6)

3. 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)



4. Filtret nedan skall implementeras med en arkitektur som använder delat minne. Beräkningselementen är av två typer, multiplikator och adderare, och var för sig homogena och non-preemptive. Latencyn är fyra tidsenheter för multiplikationerna och en tidsenhet för additioner.
The filter below is to be implemented using a shared memory architecture. The processing elements are of two types, multipliers and adders, each type is homogenous and non-preemptive. The latency is four time units for the multiplications and one time unit for the additions.



- Beräkna minimala sampelperioden 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)
- Schemulera algoritmen så att $T_{sample} = T_{min}$.
Schedule the algorithm so that $T_{sample} = T_{min}$. (8)
- Bestäm det teoretiska antalet multiplikatorer och adderare som behövs för $T_{sample} = T_{min}$.
Determine the theoretical number of multipliers and adders required for $T_{sample} = T_{min}$. (2)

5. Filtret i uppgift 4 skall implementeras med distribuerad aritmetik. Koefficienterna har följande värden: $a = b = 0.375$, $c = 0.6875$ och $d = -0.625$. Minneskodning behöver **inte** användas för att minska storleken hos ROM. *The filter in problem 4 is to be implemented using distributed arithmetic. The coefficients have the following values: $a = b = 0.375$, $c = 0.6875$, and $d = -0.625$. 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) Utgående från det ursprungliga signalflödesschemat, bestäm den minimala sampelperioden när bit-seriella operatorer används. Antag model 0, dvs inga extra register efter operatorerna.
Based on the original signal flow graph, determine the minimal sample period using bit-serial operation. Assume model 0, i.e., no extra register after each operator. (2)
- f) Utgående från ett signalflödesschema med distribuerad aritmetikenheter (bortse alltså från arkitekturen i b)), bestäm den minimala sampelperioden.
Based on a signal flow graph using distributed arithmetic (disregard the architecture in b)), determine the minimal sample period. (4)
- g) Rita ett schema för de två multiplikationerna c och d genom att använda en förenklad bitseriell/parallell multiplikator baserad på CSD representation. Använd byggblock som fulladderare, D-vippor och logiska grindar.
Draw the schematic for the two multiplications c and d by using a simplified bit-serial/parallel multiplier based on CSD representation. Use building blocks such as fulladders, flip-flops, and logic gates. (4)