20/20

1.3 The speedup of using n processors over the use of one processor in solving a computing problem was analyzed in Section 1.3.4. under various assumptions, such as $f_i = 1/n$ and $d_i = 1/i$ for $i = 1, 2, \cdots, n$.

SOLUTION: (a) Repeat the performance speedup analysis to derive a new speedup equation (similar to Eq. 1.8) under the following new probability distributions of operating modes.

$$f_i = \frac{i}{\sum_{i=1}^{n} i} \quad \text{for } i = 1, 2, \cdots, n$$

$$T_1 = \sum_{i=1}^{1} f_i d_i = \frac{1}{1} \cdot \frac{1}{1} = 1$$

$$T_n = \sum_{i=1}^{n} f_i \cdot d_i = \sum_{i=1}^{n} \frac{i}{\sum_{i=1}^{n} i} \cdot \frac{1}{i} = \frac{1}{\sum_{i=1}^{n} i} \cdot \sum_{i=1}^{n} 1$$

$$= \frac{2}{n(n+1)} \cdot n = \frac{2}{n+1}$$

$$S \text{ (speedup)} = \frac{T_1}{T_n} = \frac{n+1}{2}$$

(b) Repeat part (a) for another probability distribution:

$$f_i = \frac{n-i+1}{\sum_{i=1}^{n} i} \quad \text{for } i = 1, 2, \cdots, n$$

$$f_i = \frac{n-i+1}{\sum_{i=1}^{n} i} = \frac{2(n-i+1)}{n(n+1)}$$

$$T_1 = f_1 = \frac{2(1-1+1)}{1(1+1)} = 1$$

$$T_n = \sum_{i=1}^{n} f_i \cdot d_i = \frac{2}{n(n+1)} \sum_{i=1}^{n} \frac{n-i+1}{i}$$

$$= \frac{2}{n(n+1)} \cdot \left[(n+1) \cdot \sum_{i=1}^{n} \frac{1}{i} - n \right]$$

$$= \frac{2}{n} \sum_{i=1}^{n} \frac{1}{i} - \frac{2}{n+1}$$

$$S \text{ (speedup)} = \frac{1}{\frac{2}{n} \sum_{i=1}^{n} \frac{1}{i} - \frac{2}{n+1}}$$

2.1 Consider a two-level memory hierarchy (M_1, M_2) for a computer system, as depicted in the following diagram. Let C_1 and C_2 be the costs per bit, S_1 and S_2 be the storage capacities, and t_1 and t_2 be the access times of the memories M_1 and M_2, respectively. The hit ratio H is defined as the probability that a logical address generated by the CPU refers to information stored in M_1. Answer the following questions associated with this virtual memory system.

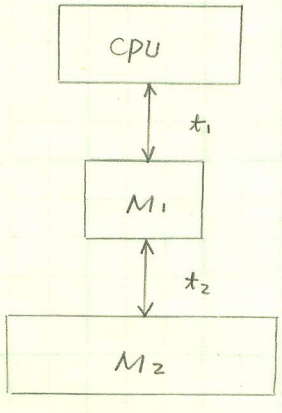
(a) What is the average cost C per bit of the entire memory hierarchy?

(b) Under what condition will the average cost per bit C approach C_2?

(c) What's the avarage access time t_a for the CPU to access a word from the memory system?

(d) Let $r = t_2/t_1$ be the speed ratio of the two memories. Let $E = t_1/t_a$ be the access efficiency of the virtual memory system. Express E in terms of r and H. Also plot E against H for $r = 1, 2, 10$ and 100 respectively on a grid-graph paper.

(e) Suppose that $r = 100$, what is the required minimum value of the hit ratio to make $E > 0.90$?



SOLUTION FOR 2.1 :

(a) THE AVERAGE COST PER BIT :

$$C = \frac{C_1 S_1 + C_2 S_2}{S_1 + S_2} = C_2 + (C_1 - C_2) \cdot \frac{1}{1 + \frac{S_2}{S_1}} \quad \text{------} \quad (1)$$

(b) FROM THE EQUATION (1) ABOVE, IF $C_1 = C_2$, THEN $C = C_2$; AND IF $S_2 \gg S_1$, WE HAVE $C \longrightarrow C_2$. SO, THE CONDITIONS ARE

(1) $C_1 \approx C_2$ OR (2) $S_2 >> S_1$ OR (3) BOTH

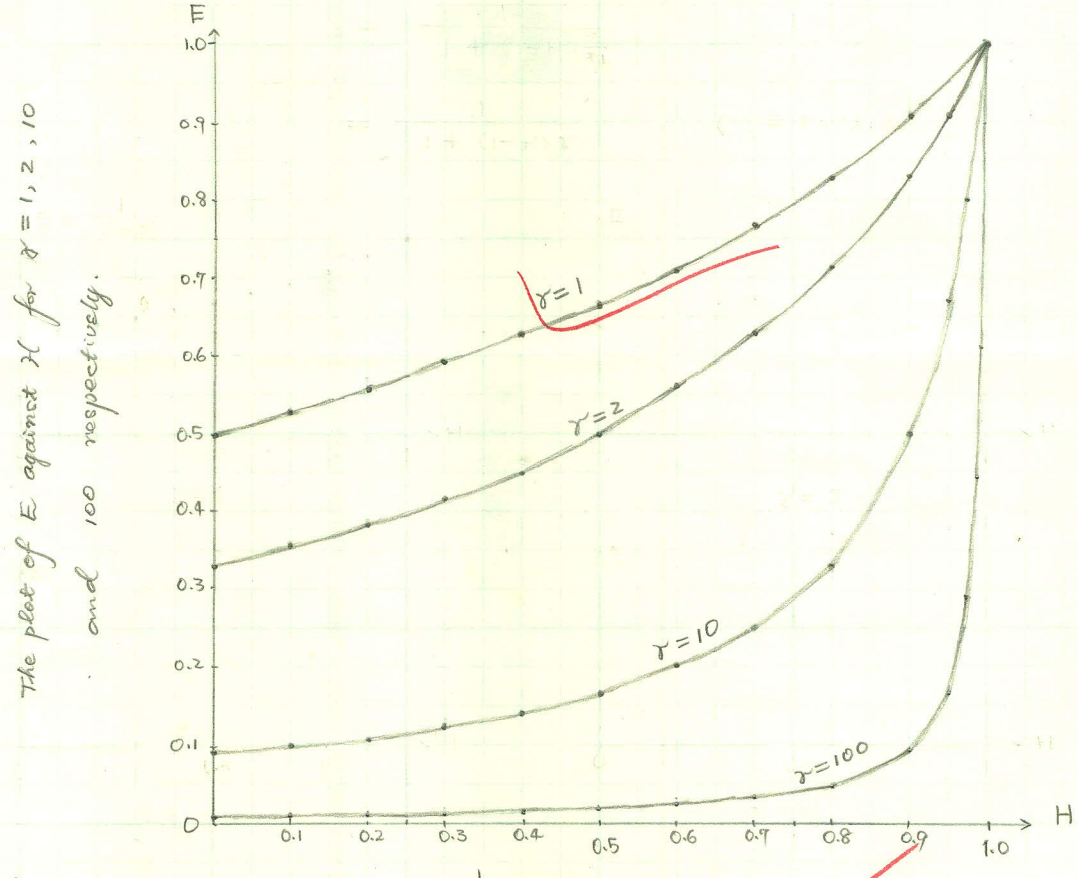
(c) THE AVERAGE ACCESS TIME :

$$t_a = H \cdot t_1 + (1-H)(t_1 + t_2) = t_1 + t_2 - H \cdot t_2$$

(d) THE ACCESS EFFICIENCY :

$$E \triangleq t_1 / t_a = t_1 / [H \cdot t_1 + (1-H)(t_1 + t_2)]$$

$$= 1 / [1 + (1-H)\frac{t_2}{t_1}] = \frac{1}{1 + (1-H)\gamma} \quad (\gamma \triangleq t_2 / t_1)$$



The plot of E against H for $\gamma = 1, 2, 10$ and 100 respectively.

(e) $\gamma = 100$; SO, $E = \frac{1}{1 + (1-H) \times 100}$

$$H = 1 - \frac{1/E - 1}{100} > 1 - \frac{1/0.90 - 1}{100} = 0.99889 \quad \leftarrow (\text{THE MIN. OF HIT RATIO})$$

17/20

EE524 Computer Architecture And

Parallel Processing

Homework 2

Benmei Chen

Gonzaga University, Spokane

Feb 11, 1987

2.5 A computer architect is considering the adoption of write-through-with-write-allocate(WTWA) or write-back cache management strategy. Assuming no read-throught, each block consists of b words, which can be transferred between main memory(MM) and cache in b+c-1 time units, where c is the MM cycle time. The cache has a hit ratio indicated by the parameter h. The probability that a memory reference is a write is wt and the probability that the block being replaced in the cache was modified(in WB strategy) is w_b. Usually w_b w_t.

a) Using each the strategy, give a formula for the expected time to process a reference in terms of the above variables.

SOLUTION: Assume the cache cycle time is t_c. And because the system above is no read-throught, we can get the equation 1 below by refering to Eq. 2.27 on page 114.

$$t_{WTWA} = t_c + (1-h)(b+c-1) + w_t*c \quad \text{..... 1}$$

From the equation 2.31 on page 115, we have

$$t_{WB} = t_c + (1-h)(1+w_b)(b+c-1) \quad \text{..... 2}$$

b) Assuming w_t=0.16 and w_b=0.56, what is the performance of the WB strtegy in comparison to WTWA strategy when 1) h → 1 and 2) h → 0.

SOLUTION: 1) In case of h → 1 : From the equations 1 and 2 above, we get

$$t_{WTWA} = t_c + w_t*c > t_{WB} = t_c;\ t_{WTWA}-t_{WB}=w_t*c=0.16*c>0.$$

2) In case of h → 0: From the equations, we have

$$t_{WTWA} = t_c + (b+c-1) + w_t*c < t_{WB}=t_c+(b+c-1)+w_b*(b+c-1)$$

Because,

$$t_{WTWA} - t_{WB} = (w_t-w_b)*c - w_b*(b-1)$$
$$= (0.16 - 0.56)*c - 0.56*(b-1)$$
$$= -0.40*c-0.56*(b-1) < 0$$

c) Give a general expression describing when WTWA is better than WB as a function of h and b. Assume that w_t=0.16, w_b=0.56, and c=10.

SOLUTION: From the equations 1 and 2, and let w_t=.16, w_b=.56 and c=10, we obtain, (please see page 5 for the plot)

$$t_{WB} - t_{WTWA} = w_b*(1-h)*(b+c-1) - w_t*c$$
$$= 0.56*(1-h)*(b+9) - 0.16*10$$
$$= 3.44 + 0.56*(b-h*b-9h) \quad \text{..... 3}$$

So, WTWA is better than WB when $3.44+.56*(b-h*b-9h)>0$.

d) Does w_t depend on h ? Give intuitive reasons.

SOLUTION: NO. It only depends on how many memory references are write and how often the references are write operations.

2.6 A certain uniprocessor computer system has a paged segmentation virtual memory system and also a cache. The virtual address is a triple (s,p,d) where s is the segment number, p is the page within s, and d is the displacement within p. A translation lookaside buffer (TLB) is used to perform the address translation when the virtual address is in the TLB. If there is a miss in the TLB, the translation is performed by accessing the segment table and then the page table, either or both of which may be in the cache or in main memory (MM).

Address translation via the TLB requires one clock cycle. A fetch from the cache requires two clock cycles(one clock cycle to determine if the requested address is in the cache plus one clock cycle to read the data). A read from MM requires eight clock cycles. There is no overlap between TLB translation and cache access. Once the address translation is complete, the read of the desired data may be from either the cache or MM. This means that the fastest possible data access requires three clock cycles: one for TLB address translation and two to read the data from the cache. Threr are nine other ways in which a read can proceed, all requiring more than three clock cycles.

a) Assuming a TLB hit ratio of 0.9 and a cache hit ratio of h, enumerate all 10 possible read patterns, the time taken for each, and the probability of occurrence for each pattern. What is the average read time in the system?(Assume that when a word is fetched from memory, a read-through policy is used.)

SOLUTION TO THIS PART IS ON PAGE 3&4.

b) The above discussion assumes that the cache is always given a physical memory address. Suppose that the cache is presented with the virtual address of the data being requested rather than its physical address in memory. In this case, the TLB translation and cache search can be done concurrently. This means that whenever the requested data is in the cache, no address translation is necessary and only two clock cycles are required for the fetch. If the data is not in the cache, either a TLB translation segment table-page table access is needed to generate the physical address of the data. When data is written into the cache, it is tagged with its virtual address. Find the average read time for a system organized in this fashion. Assume that only one clock cycle is required to establish that an item is not in the cache.

SOLUTION TO THIS PART IS ON PAGE 4.

c) What are the disadvantages of a cache using virtual address?

SOLUTION TO THIS PART IS ON PAGE 4.

SOLUTION TO PROBLEM 2.6 PART a:

Pattern 1 : The virtual address is in the TLB and the data is in the cache.

Pattern 2 : The virtual address is in the TLB and the data is in the MM.

Pattern 3 : The virtual address is not in TLB, and the segment table is in the cache, and the page table as well as the data are in the cache too.

Pattern 4 : The virtual address is not in the TLB. But, the segment table and the page table are in the cache. The data, however, is not in the cache, but the MM.

Pattern 5 : The virtual address is not in the TLB. The segment table is in the cache. The page table is not in the cache. And the data is in the cache.

Pattern 6 : The virtual address is not in the TLB. The segment table is in the cache. The page table is not in the cache, but in the MM. The data is in the MM too.

Pattern 7 : The virtual address is not in the TLB. The segment table is not in the cache, but in the MM. And the page table and the data are in the cache.

Pattern 8 : The virtual address is not in the TLB. The segment table is in the MM. The page table is in the cache. But, the data is in the MM.

Pattern 9 : The virtual address is not in the TLB. The segment table is not in the cache. The page table is not in the cache too. Both tables are in the MM. THE data is in the cache.

Pattern 10: The virtual address is not in the TLB. The segment table is in the MM. And both the page table and data are in the main memory.

The time taken for each pattern:(the system uses the read-throught policy)

Pattern 1 : 3 clock cycles= 1 clock cycle + 2 clock cycles
Pattern 2 : 9 clock cycles= 1 clock cycle + 8 clock cycles
Pattern 3 : 7 clock cycles= 1 + 2 + 2 + 2 clock cycles
Pattern 4 : 13 clock cycles=1 + 2 + 2 + 8 clock cycles
Pattern 5 : 13 clock cycles=1 + 2 + 8 + 2 clock cycles
Pattern 6 : 19 clock cycles=1 + 2 + 8 + 8 clock cycles
Pattern 7 : 13 clock cycles=1 + 8 + 2 + 2 clock cycles
Pattern 8 : 19 clock cycles=1 + 8 + 2 + 8 clock cycles
Pattern 9 : 19 clock cycles=1 + 8 + 8 + 2 clock cycles
Pattern 10: 25 clock cycles=1 + 8 + 8 + 8 clock cycles

TO BE CONTINUED ON NEXT PAGE.

The Probability of Occurrence for Each Pattern:

Pattern 1 : 0.9*h
Pattern 2 : 0.9*(1-h)
Pattern 3 : 0.1*h*h*h
Pattern 4 : 0.1*h*h*(1-h)
Pattern 5 : 0.1*h*(1-h)*h
Pattern 6 : 0.1*h*(1-h)*(1-h)
Pattern 7 : 0.1*(1-h)*h*h
Pattern 8 : 0.1*(1-h)*h*(1-h)
Pattern 9 : 0.1*(1-h)*(1-h)*h
Pattern 10: 0.1*(1-h)*(1-h)*(1-h)

The Average Time For Reading :

$T_1 = 3*.9*h + 9*.9*(1-h) + 7*.1*h*h*h + 13*.1*h*h*(1-h)$

$+ 13*.1*h*(1-h)*h + 19*.1*h*(1-h)*(1-h) + 13*.1*(1-h)*h*h$

$+ 19*.1*(1-h)*h*(1-h) + 19*.1*(1-h)*(1-h)*h + 25*.1*(1-h)^3$

$= 10.6 - 7.2*h$ clock cycles

* SOLUTION TO PROBEM 2.6 PART b :

The read time for each pattern in a cache using virtual address:

Pattern 1 : 2 clock cycles = 0 + 2 + 0 clock cycles
Pattern 2 : 10 clock cycles = 1 + 8 + 1 clock cycles
Pattern 3 : 2 clock cycles = 0 + 0 + 0 + 2 clock cycles
Pattern 4 : 14 clock cycles = 1 + 2 + 2 + 8 + 1 clock cycles
Pattern 5 : 2 clock cycles
Pattern 6 : 20 clock cycles = 1 + 2 + 8 + 8 + 1 clock cycles
pattern 7 : 2 clock cycles =
Pattern 8 : 20 clock cycles = 1 + 8 + 2 + 8 + 1 clock cycles
Pattern 9 : 2 clock cycles
Pattern 10: 26 clock cycles = 1 + 8 + 8 + 8 + 1 clock cycles

The Average read time for this system:

$T_2 = 2*.9*h + 10*.9*(1-h) + 2*.1*H*h*h + 14*.1*h*h*(1-h)$

$+ 2*.1*h*(1-h)*h + 20*.1*h*(1-h)*(1-h) + 2*.1*(1-h)*h*h$

$+ 20*(1-h)*h*(1-h) + 2*.1*(1-h)*(1-h)*h + 26*(1-h)^3*.1$

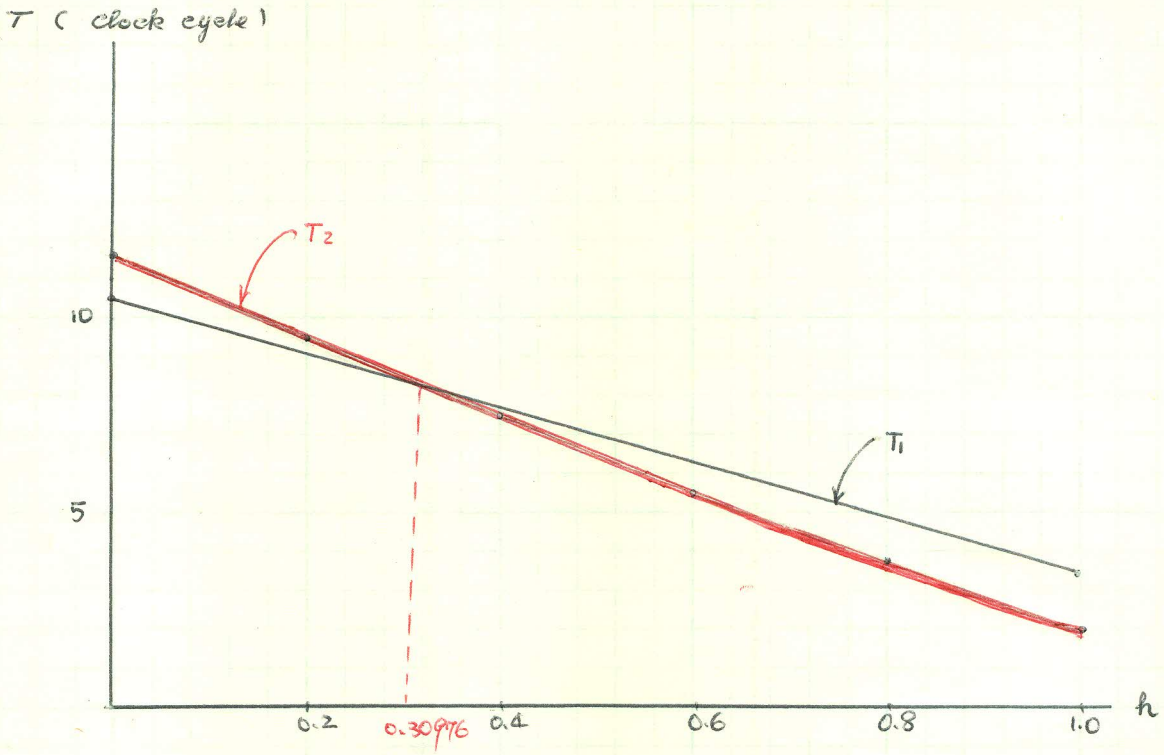
$= 1.2*h^2 - 10.8*h + 11.6$ clock cycles

* SOLUTION TO PROBLEM 2.6 PART c :

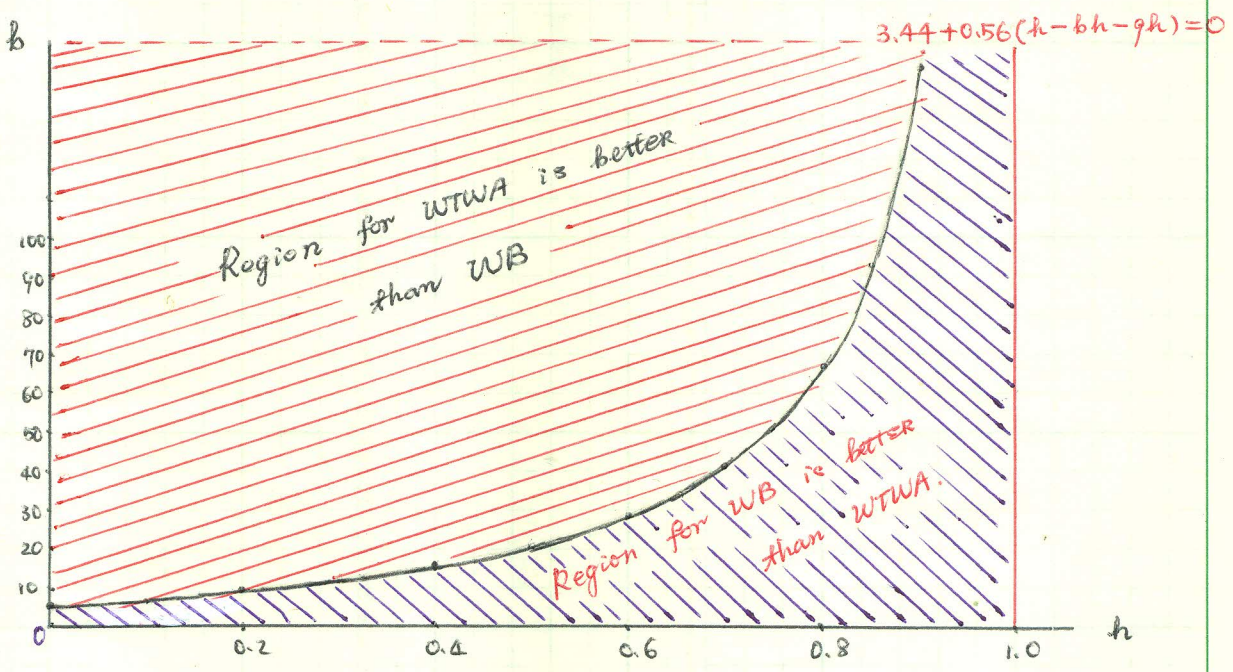
For the cache using virtual address, there are two disadvantages:

1) Need more space in cache to to fit the virtual addresses.
   It costs a lot.

2) From the plots below , we can find out that the fashion is even worse than the one in part (a) when $h < 0.30976$.



The plots for reading times in part a and b. of prob. 2.6



The plot of relationship between b. h and

t_{WB} . t_{WTWA} in Prob. 2.5 point e.

3.2 Compare the advantages and disadvantages of the three interleaved memory organizations: the S-access, the C-acess, and the C/S-access described in Section 3.14 for pipelined vector accessing. In the comparison, you should be concerned with the issues on effective memory bandwidth, storage schemes used, access conflict resolution, and cost-effectiveness tradeoffs.

S-access: Accessing all modules simultaneously.

C-access: Accessing modules concurrently.

C/S-access: h different accesses to blocks of M consecutive words can be in progress simultaneously.

Assume the memory access time is Ta, and the address sequence is generated with a skip distance d.

COMPARISON:

(1) Memory Bandwidth (MB).

i) S-access has an average data rate of dTa/M when $d \leq M$, and Ta, when $d > M$

So, $MB_s = M/dTa$ WHEN $d \leq M$, and $1/Ta$, when $d > M$

ii) C-access has an max. rate of Ta/M per word. So, $MB_c = M/Ta$.

Obviously, WHEN $d > 1$, $MB_s < MB_c$

iii) C/S-access has a memory bandwidth between these two.

(2) Access Conflict Resolution.

i) In the S access scheme, if a conflict occurs, it takes Ta to wait the completation of former access.

ii) In the C access scheme, only needs Ta/M.

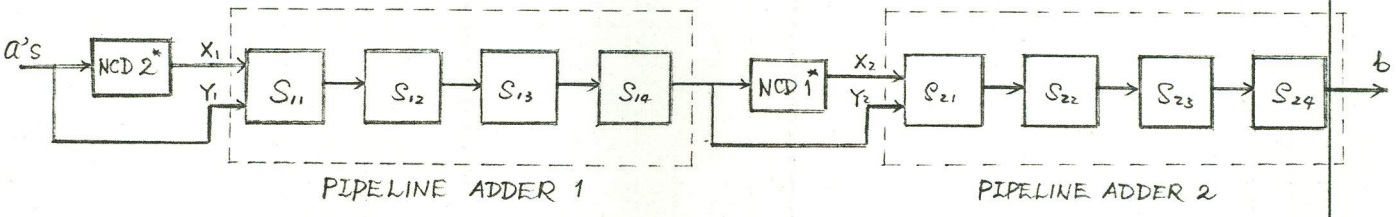
iii) C/S-access is between these two.

(3) Cost-effetiveness.

C-access needs an address latch, increasing the cost. C/S-access isn't easy to control.

3.8 (a) Suppose that only two 4-segment pipelined adders and a number of noncompute delay elements are available. The delay of each segment is one time unit and the noncompute delay element can have either a one- or two-time unit delay. Using available resources, construct a pipeline with only one input, a's, to compute $b(i) = a(i) + a(i-1) + a(i-2) + a(i-3)$. Show the schematic block diagram of your design.

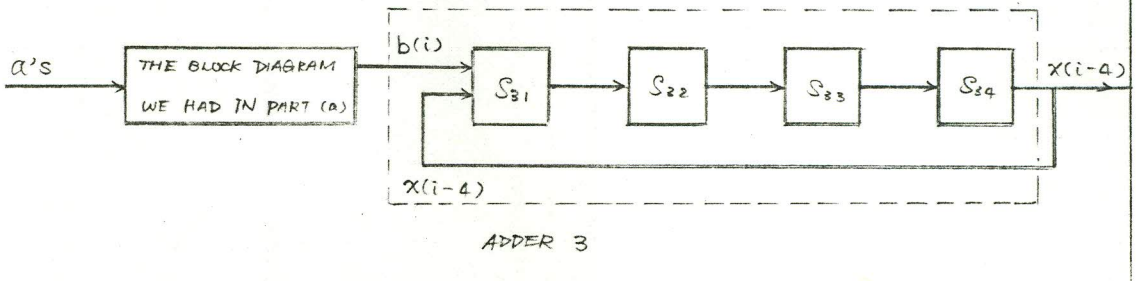
The following is the schematic block diagram of my design:



* NCD STANDS FOR Non-Compute Delay

SHOW: Assume that at a certain time, we have a's = a(i). Then, at the input of the first adder, we have $X_1 = a(i-2)$, $Y_1 = a(i)$. After four compute delay in the segments of pipeline adder, we have $Y_2 = a(i) + a(i-2)$ and $X_2 = a(i-1) + a(i-3)$. At last, we have $b(i) = a(i) + a(i-1) + a(i-2) + a(i-3)$ After another four time units delay in the adder 2.

(b) Given one additional four-segment pipelined adder, use this adder together with the pipeline obtain from (a) to design a pipeline for computing the recurrence function $x(i) = a(i) + x(i-1)$. The pipeline constructed should have a feedback. Show your schematic block diagram. Hint: $x(i) = a(i) + x(i-1) = a(i) + [a(i-1) + x(i-2)] = a(i) + a(i-1) + [a(i-2) + x(i-3)] = a(i) + a(i-1) + a(i-2) + [a(i-3) + x(i-4)] = b(i) + x(i-4)$



ADDER 3

3.5 For the following reservation table of a pipeline processor, give the forbidden list of avoided latencies F, the lower bound on latency, the collision vector, the state diagram, the MAL and all greedy cycles.

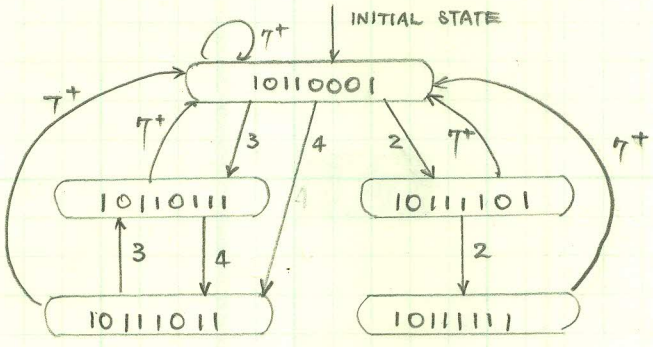
18/20

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
S_1	X								X
S_2		X	X					X	
S_3				X					
S_4				X	X				
S_5						X	X		

SOLUTION: THE FORBIDDEN SET OF THE LATENCY $F = \{1, 5, 6, 8\}$

THE COLLISION VECTOR : $C = \{\ 10110001\ \}$

THE STATE DIAGRAM :

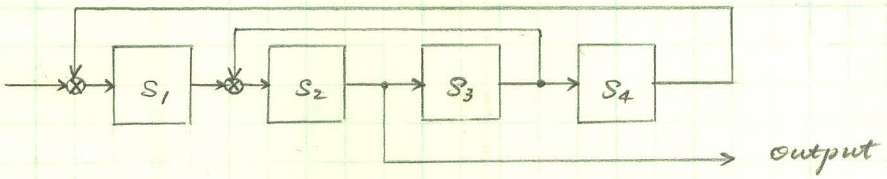


THE MAL = (3+4)/2 = 3.5

THE GREEDY CYCLES : (3, 4) and (2, 2, 7)

THE LOWER BOUND ON LATENCY : 2

3.9. Consider the following pipelined processor with four stages. All successor stages after each stage must be used in successive clock periods.



Answer the following questions associated with using this pipeline with an evaluation time of six pipeline clock periods.

(a) Write out the reservation table for this pipeline with six columns and four rows.

	t_0	t_1	t_2	t_3	t_4	t_5
S_1	X				X	
S_2		X		X		X
S_3			X		X	
S_4				X		X

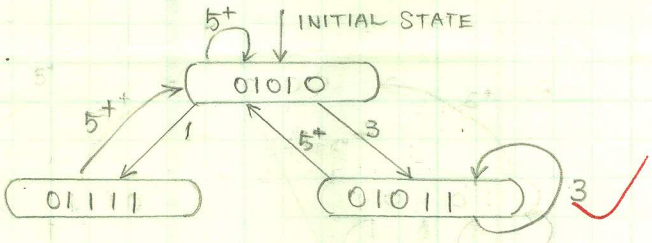
(b) List the set of forbidden latencies between task initiations.

$F = \{2, 4\}$ ✓

(c) Show the initial collision vector.

$C = (\ 01010\ )$ ✓

(d) Draw the state diagram which shows all the possible latency cycles.



QUESTION: According to the statement on page 205: The shift register will be set to the initial state, if the latency (shift) is greater than or equal to n. But, the example in the text sets the shift register to the initial state only after 7 latency shift, which less than n = 8. Why?

This result follows algorithm given in class.

(e) List all the simple cycles from the state diagram.

(f) List all the greedy cycles from the state diagram.

Simple cycles	Average latency
(5)	4
(1, 5)*	3
(3, 5)	4
(3)*	3

* Greedy cycles

(g) What is the value of the minimal average latency (MAL)?

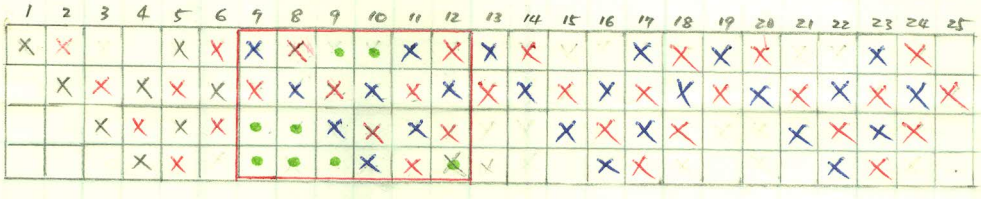
MAL = 3

(h) Indicate the minimum constant latency cycles for this pipeline.

(3)

(i) What is the maximal throughput of this pipeline?

The maximal throughput only happens when we use the MAL. From the table below, we have



Assume we have n tasks, then

$$E = \frac{\text{Busy time span}}{\text{whole time span}} = \frac{\text{Total (X)}}{\text{Total (X)} + \text{Total (•)}} = \frac{16}{24} = \frac{2}{3}$$

$E_{MAL} = 0.333\ldots$

Maximal throughput $\approx E/\tau = \frac{2}{3\tau}$, where τ is clock period.

3.11 Answer the following questions related to the task initiation cycle (2,3,7) for a given pipelined processor.

(a) What are the period p and the average latency l_a of this initiation cycle ?

$p = 2+3+7 = 12$; $l_a = (2+3+7)/3 = 4$ ✓

(b) Specify the initiation interval set $G \pmod p$.

$G = \{2, 3, 5, 7, 9, 10, 12, 14, 15, 17, \cdots\}$

$G \pmod{12} = \{0, 2, 3, 5, 7, 9, 10\}$ ✓

(c) What is the necessary and sufficient condition that a given task initiation cycle is allowed by a pipline with a forbidden latency set F ? Repeat the same question for a constant initiation cycle with period p.

i) IN THIS PARTICULAR QUESTION, WE HAVE $G \pmod{12} = \{0,2,3,5,7,9,10\}$ SO, $\overline{G} \pmod{12} = \{1,4,6,8,11\}$. THEREFORE, THE GIVEN TASK INITIATION CYCLE IS ALLOWED BY A PIPELINE, IF, AND ONLY IF F EQUAL TO ANY SUBSET OF $\{1,4,6,8,11\}$. ✓

ii) $p = 12$; $C' = (12)$

$G_{C'} = \{12, 24, 36, \cdots\}$

$G_{C'} \pmod{12} = \{0\}$ ✗ -1

$\overline{G}_{C'} \pmod{12} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$

SO, F CAN BE CHOICED ANY SUBSET OF $\{1, 2, 3, \cdots, 10, 11\}$