

5.4. (a) It takes $5 + 8 = 13$ clock cycles.

$$\text{Total time} = \frac{13}{(133 \times 10^6)} = 0.098 \times 10^{-6} \text{ s} = 98 \text{ ns}$$

$$\text{Latency} = \frac{5}{(133 \times 10^6)} = 0.038 \times 10^{-6} \text{ s} = 38 \text{ ns}$$

(b) It takes twice as long to transfer 64 bytes, because two independent 32-byte transfers have to be made. The latency is the same, i.e. 38 ns.

1

[Home](#) | [Contact](#) | [DMCA](#)

Book Descriptions:

computer organization hamacher instructor manual solution

Same as 69 in b 7.6. Many approaches are possible. For example, the three machine instructions implemented by the control sequences in parts a, b, and c can be thought of as one instruction, Add, that has three addressing modes, Immediate Imm, Absolute Abs, and Indirect Ind, respectively. In order to simplify the decoder block, hardware may be added to enable the control step counter to be conditionally loaded with an outofsequence number at any time. This provides a "branching" facility in the control sequence. The three control sequences may now be merged into one, as follows 14. Same as in a 5. Zout, PCin, If Imm branch to 10 1 6. WMFC 7. MDRout, MARin, Read, If Abs branch to 10 8. WMFC 9. MDRout, MARin, Read 10. R1out, Yin, WMFC 11. MDRout, Add, Zin 12. Zout, R1in, End Depending on the details of hardware timing, steps, 6 and 7 may be combined. Similarly, steps 8 and 9 may be combined. 7.7. Following the timing model of Figure 7.5, steps 2 and 5 take 16 ns each. Assume that the hardware is configured such that register Z can perform the function of the accumulator, register TEMP can be used to hold the multiplier and is connected to register Z for shifting as shown. Register Y will be used to hold the multiplicand. Furthermore, the multiplexer at the input of the ALU has three inputs, 0, 4, and Y. To simplify counting, a counter register is available on the bus. It is decremented by a control signal Decrement and it sets an output signal Zero to 1 when it contains zero. A facility to place a constant value on the bus is also available. For example, it cannot be used to implement register Z in Figure 7.1. It may be used in other registers, provided that hold time requirements are met. 7.14. The presence of a gate at the clock input of a flipflop introduces clock skew. This means that clock edges do not reach all flipflops at the same time. For example, consider two flipflops A and B, with output QA connected to input DB. <http://www.nutstudio.it/userfiles/870-express-super-magnum-manual.xml>

• computer organization hamacher instructor manual solution, computer organization hamacher instructor manual solutions.

A clock edge loads new data into A, and the next clock edge transfers these data to B. However, if clock B is delayed, the new data loaded into A may reach B before the clock and be loaded into B one clock period too early. QA QB ClockA ClockB ClockA QA ClockB skew In the absence of clock skew, flipflop B records a 0 at the first clock edge. However, if Clock B is delayed as shown, the flipflop records a 1. 3 7.15. Add a latch similar to that in Figure A.27 at each of the two register file outputs. A read operation is performed in the RAM in the first half of a clock cycle and the latch inputs are enabled at that time. The data read enter the two latches and appear on the two buses immediately. During the second phase of the clock the latch inputs are disabled, locking the data in. Hence, the data read will continue to be available on the buses even if the outputs of the RAM change. The RAM performs a write operation during this phase to record the results of the data transfer. Bus A Bus B Bus C RAM Read Write Enablein Clock Read Write Enablein 7.16. The step counter advances at the end of a clock period in which Run is equal to 1. With reference to Figure 7.5, Run should be set to 0 during the first clock cycle of step 2 and set to 1 as soon as MFC is received. In general, Run should be set to 0 by WMFC and returned to 1 when MFC is received. To account for the possibility that a memory operation may have been already completed by the time WMFC is issued, Run should be set to 0 only if the requested memory operation is still in progress. A state machine that controls bus operation and generates the run signal is given below. For clarity, we have assumed that MDR consists of two separate registers for input and output data. Multiplexers Mux1 and Mux2 select input B for even and input A for odd byte operations. Mux 3 selects input A for word operations and input B for byte operations. Input B provides either zero extension or sign extension of byte

operands. <http://brod-plovdiv.com/images/870-manual-remington.xml>

For signextension it should be connected to the most significant bit output of multiplexer Mux2. Memory bus MDRH in MDRL in MDRH out MDRL out Zero or Sign ext. B A Mux 3 B A Mux 1 Mux 2 B A 7.19. Use the delay element in a ring oscillator as shown below. By adding the control circuit shown, the oscillator will run only while Run is equal to 1. When stopped, its output A is equal to 0. The oscillator will always generate complete output pulses. When this line changes to 0, Enable changes to 0 for one clock cycle. The feed back circuit has been modified to make the counter selfstarting. The advantage of a Johnson counter is that there are no glitches in decoding the count value to generate the timing signals. D Q D Q D Q T5 T6 T7 T0 T1 T2 D Q D T8 T3 Q T9 T4 7.22. We will generate a signal called Store to recirculate data when no external action is required. This is a Moore machine. The output values are given inside each state as they are functions of the state only. Since there are 6 independent states, a minimum of three flipflops r, s, and t are required for the implementation. A possible state assignment is shown in the diagram. Discover everything Scribd has to offer, including books and audiobooks from major publishers. Start Free Trial Cancel anytime. Report this Document Download Now Save Save Computer Organization Hamacher Instructor Manual S. For Later 0 ratings 0% found this document useful 0 votes 398 views 3 pages Computer Organization Hamacher Instructor Manual Solution Chapter 1 Uploaded by Milos Stojanovic Description organization Full description Save Save Computer Organization Hamacher Instructor Manual S. For Later 0% 0% found this document useful, Mark this document as useful 0% 0% found this document not useful, Mark this document as not useful Embed Share Print Download Now Jump to Page You are on page 1 of 3 Search inside document. Discover everything Scribd has to offer, including books and audiobooks from major publishers.

Start Free Trial Cancel anytime. Report this Document Download Now Save Save Computer Organization Hamacher Instructor Manual S. For Later 0 ratings 0% found this document useful 0 votes 422 views 13 pages Computer Organization Hamacher Instructor Manual Solution Chapter 4 Uploaded by suduku007 Description Computer Organization Hamacher Instructor Manual Solution Chapter 4 Full description Save Save Computer Organization Hamacher Instructor Manual S. For Later 0% 0% found this document useful, Mark this document as useful 0% 0% found this document not useful, Mark this document as not useful Embed Share Print Download Now Jump to Page You are on page 1 of 13 Search inside document. Download Computer Organization Hamacher Instructor Manual Solution Chapter 2. Byte 1007 shown as XX is unchanged. See Section 2.6.3 for hex notation. 2.6. Byte contents in hex, starting at location 1000, will be 4A, 6F, 68, 6E, 73, 6F, 6E. The two words at 1000 and 1004 will be 6E686F4A and XX6E6F73. The Base with index addressing mode R1,R2 is used to access the scores on a particular test. Register R1 points to the test score for student 1, and R2 is incremented by Stride in the inner loop to access scores on the same test by successive students in the list. Initialize base register R1 to the location of the test 1 score for student 1. Initialize register R3 to the location of the sum for test 1. Initialize outer loop counter R10 to j. Initialize inner loop counter R11 to n. Clear index register R2 to zero. Clear sum register R0 to zero. Accumulate the sum of test scores in R0. Increment index register R2 by Stride value. Check if all student scores on current test have been accumulated. Store sum of current test scores and increment sum location pointer. Instead of using R0 to accumulate the sum, the sum can be accumulated directly into DOTPROD.

This means that the last Move instruction in the program can be removed, but DOTPROD is read and written on each pass through the loop, significantly increasing memory accesses. The four registers R1, R2, R3, and R4, are still needed to make this program efficient, and they are all used in the loop. Suppose that R1 and R2 are retained as pointers to the A and B vectors. Use R4 to contain distance in bytes Stride between successive elements in a column. Byte distances from A0,0 to A0,x and A0,y placed in R1 and R2. Add corresponding column elements. Increment column element pointers by

Stride value. Repeat until all elements are added. Restore R5. Return to calling program. 2.16. The assembler directives ORIGIN and DATAWORD cause the object program memory image constructed by the assembler to indicate that 300 is to be placed at memory word location 1000 at the time the program is loaded into memory prior to execution. That is, the next item must be entered at the beginning of the memory region, assuming that location is empty. b A current queue of bytes is shown in the memory region from byte location 1 to byte location k in the following diagram. Increasing addresses Current queue of bytes 1 k.. OUT IN The IN pointer points to the location where the next byte will be appended to the queue. If the queue is not full with k bytes, this location is empty, as shown in the diagram. The OUT pointer points to the location containing the next byte to be removed from the queue. If the queue is not empty, this location contains a valid byte, as shown in the diagram. Initially, the queue is empty and both IN and OUT point to location 1. c Initially, as stated in Part b, when the queue is empty, both the IN and OUT pointers point to location 1. When the queue has been filled with k bytes and none of them have been removed, the OUT pointer still points to location 1.

<https://cohemployeenews.com/images/canada-post-supervisor-manual-and-mechanized.pdf>

But the IN pointer must also be pointing to location 1, because following the wraparound rule it must point to the location where the next byte will be appended. Thus, in both cases, both pointers point to location 1; but in one case the queue is empty, and in the other case it is full. d One way to resolve the problem in Part c is to maintain at least one empty location at all times. If the ID of the new record matches the ID of a later record in the current list, the new record will be inserted immediately after that record, including the case where the matching record is the Tail record. In this latter case, the new record becomes the new Tail record. We are a nonprofit group that run this website to share documents. We need your help to maintenance this website. Hence, the total time needed is 7 cycles. 7.3. Steps 2 and 5 will take 2 cycles each. Same as 69 in b 7.6. Many approaches are possible. For example, the three machine instructions implemented by the control sequences in parts a, b, and c can be thought of as one instruction, Add, that has three addressing modes, Immediate Imm, Absolute Abs, and Indirect Ind, respectively. In order to simplify the decoder block, hardware may be added to enable the control step counter to be conditionally loaded with an outofsequence number at any time. This provides a branching facility in the control sequence. The three control sequences may now be merged into one, as follows 14. Same as in a 5. Zout, PCin, If Imm branch to 10 1 6. WMFC 7. MDRout, MARin, Read, If Abs branch to 10 8. WMFC 9. MDRout, MARin, Read 10. R1out, Yin, WMFC 11. MDRout, Add, Zin 12. Zout, R1in, End Depending on the details of hardware timing, steps, 6 and 7 may be combined. Similarly, steps 8 and 9 may be combined. 7.7. Following the timing model of Figure 7.5, steps 2 and 5 take 16 ns each.

<http://columbuscigar.com/images/canada-manual-transmission-cars.pdf>

Assume that the hardware is configured such that register Z can perform the function of the accumulator, register TEMP can be used to hold the multiplier and is connected to register Z for shifting as shown. Register Y will be used to hold the multiplicand. Furthermore, the multiplexer at the input of the ALU has three inputs, 0, 4, and Y. To simplify counting, a counter register is available on the bus. It is decremented by a control signal Decrement and it sets an output signal Zero to 1 when it contains zero. A facility to place a constant value on the bus is also available. For example, it cannot be used to implement register Z in Figure 7.1. It may be used in other registers, provided that hold time requirements are met. 7.14. The presence of a gate at the clock input of a ipop introduces clock skew. This means that clock edges do not reach all ipops at the same time. For example, consider two ipops A and B, with output QA connected to input DB. A clock edge loads new data into A, and the next clock edge transfers these data to B. However, if clock B is delayed, the new data loaded into A may reach B before the clock and be loaded into B one clock period too early. QA QB ClockA ClockB ClockA QA ClockB skew In the absence of clock skew, ipop B records a 0

at the rst clock edge. However, if Clock B is delayed as shown, the ipop records a 1. 3 7.15. Add a latch similar to that in Figure A.27 at each of the two register le outputs. A read operation is performed in the RAM in the rst half of a clock cycle and the latch inputs are enabled at that time. The data read enter the two latches and appear on the two buses immediately. During the second phase of the clock the latch inputs are disabled, locking the data in. Hence, the data read will continue to be available on the buses even if the outputs of the RAM change. The RAM performs a write operation during this phase to record the results of the data transfer.

Bus A Bus B Bus C RAM Read Write Enablein Clock Read Write Enablein 7.16. The step counter advances at the end of a clock period in which Run is equal to 1. With reference to Figure 7.5, Run should be set to 0 during the rst clock cycle of step 2 and set to 1 as soon as MFC is received. In general, Run should be set to 0 by WMFC and returned to 1 when MFC is received. To account for the possibility that a memory operation may have been already completed by the time WMFC is issued, Run should be set to 0 only if the requested memory operation is still in progress. For clarity, we have assumed that MDR consists of two separate registers for input and output data.

Multiplexers Mux1 and Mux2 select input B for even and input A for odd byte operations. Mux 3 selects input A for word operations and input B for byte operations. Input B provides either zero extension or sign extension of byte operands. For signextension it should be connected to the mostsignificant bit output of multiplexer Mux2.Memory bus MDRH in Zero or Sign ext. B Mux 3 A B MDRL in MDRH out MDRL out A Mux 2 B Mux 1 A 7.19. Use the delay element in a ring oscillator as shown below. By adding the control circuit shown, the oscillator will run only while Run is equal to 1. When stopped, its output A is equal to 0. The oscillator will always generate complete output pulses. The feed back circuit has been modied to make the counter selfstarting. The advantage of a Johnson counter is that there are no glitches in decoding the count value to generate the timing signals. D Q D Q D Q D Q T5 T6 T7 T8 T9 T0 T1 T2 T3 T4 7.22. We will generate a signal called Store to recirculate data when no external action is required. This is a Moore machine. The output values are given inside each state as they are functions of the state only. Since there are 6 independent states, a minimum of three ipops r, s, and t are required for the implementation. A possible state assignment is shown in the diagram.

Every instruction that is executed must be fetched from the cache, and an additional fetch from the main memory must be performed for 4% of these cache accesses.