|  |  |
| --- | --- |
| **True and False**  T: A decoder decodes 0112 to 000100001  F : A multiplexer selects two data points as output  T : An input event has a higher priority over an ouput event when they are checked  T : With an adder-subtracter the first carry-in is set to 0 for adding and 1 for subtracting  T : To subtract 5 from 7, it to add 01112 with 110112  F : The highest number of T-Cycles is 16 but it is never used  T : The memory addressed by AR directly, not through the bus  F : A control signal does not need to be defined when it is not involved  F : To write to the memory, the location address is first given through the bus  F : The BC computer can handle another I/O event while currently servicing one  F : To encode for the bus control s0 = x1 + x3 +x5 + x7  T : A multiplexor selects from k Data points would need k2 select signals  : The input event is checked first before the output event in the BC  : In a 5-bit adder/subtractor : the first carry-in is set 0 to do subtraction  : To transfer from DR register to AC register, it goes through Adder Logic  T : To fetch a data, the write signal is needed  T : The Instructions ISZ requires the largest number of T Cycles.  T : A decoder decodes a binary input to a unary output.  T : A multiplexor chooses one of multiple input as its single output.  F : To program a Verilog code module that performs the boolean function AB+CD, a minimum of 5 logic-gate statements are needed.  F : If a multiplexor is given k select input signals, it can only select one output from this many input: 2 times k.  F : When input to a decoder is 1012, then the output would be 00001000. | **Multiple Choice**   1. To flip a bit or not, we use this gate logic by setting the other input 2. To get the data from memory, Si signals to the bus must be :    1. 1112 3. Compare the sizes of registers 4. Instruction fetched from location 100 gets executed as an I/O event occurs, and the CPU is now executing I/O program. What is M[] at this moment? 5. About the E indicator in the adder/subtractor it    1. Tells wether the last carry-out bit is needed 6. To perform an “add” in BC : one operand comes AC , the other from:    1. DR 7. Which is checked first when looking for any I/O events    1. IEN 8. For the memory capacity in the BC computer the PC is    1. 12 Bits 9. Instructions 0 BSA 900 would have a pairing instruction    1. 1 BUN 901 10. AC is 8E0A16, E is 1, After execution CIL, AC is :     1. 1C15 11. To fetch data from memory, the select signals to the bus are 12. The instructions fetched from location 300 is being executed as an I/O, then the CPU executing in the I/O program, What is in M[0] 13. Instructions 1 BUN 900 would pair with this instruction 14. Which below is not in a Full Adder:     1. a overflow bit 15. How many components are there in the BC CPU that write to the common bus?     1. 7 16. An "Indirect" memory-reference instruction has extra work at the clock cycle T (number?)     1. 3 |
| **Fill Out the Blank**   1. **Control Unit**  manage the CPU 2. **rB9** is the control signals to perform CPU instruction CMA 3. ------signal is given at the end of each CPU instruction execution 4. **201** Instruction 0 BSA 500 fetched from M[200] is executed. M[500] is now? 5. **T3** is the time cycle that an “indirect” treatment is being performed. 6. **PC** The ISZ instruction can skip the next instruction by increment this register 7. **----**The last CPU instruction of the I/O handling program 8. **----**This logic gate is often used to check the parity bit in data. 9. **----**Is the number of xor gates in a 5-bit adder/subtractor 10. **----**is the longest CPU instruction to execute 11. **----**As 0 BSA 700 from M[200] gets executed, M[700] would contain? 12. **----**Is the T cycle that PC register gets incremented 13. **----**In ISZ execution the next instruction is skipped as this register zero 14. **----**Is the 2nd to last instruction is the I/O handling program 15. **----**module and parity module make a full-adder module 16. from which its function is to generate (two words) \_\_Control\_\_Signals\_\_. 17. The timing device in the control unit is called (two words) \_\_Sequence\_\_Counter\_\_ 18. which it is a counter of (how many) \_\_4\_\_ binary bits, 19. and this can indicate a total of (how many) \_\_16\_\_ T-steps. 20. D0 through D7 are decoded results from this CPU register: \_\_IR\_\_. | Write Verilog modules: a 3x8 decoder and a 8x1 multiplexor. The multiplexor “includes” the decoder module as part of it. Use arrays as much as possible. Do note write the test module. |
| For these CPU instructions : AND, ADD, LDA, and BUN  Translate their microoperations into signals  For each signal, derive a concise Boolean equation list alphabetically   |  |  | | --- | --- | | ADD DIT4: Read, LD(DR), X7  DIT5: ADD, LD(e), LD(AC) CLR (SC)  AND DOT4: Read, LD(DR) x7  DOT5: AND, CLR(SC), LD(AC  BUN D4T4: CLR(SC), LD(PC), x1  LDA D2T4: read, LD(DR), x7  D2T5: LD(AC), CLR(SC) | ADD = D1T5  AND = D0T5  CLR(SC) = D1T5 + D0T5 + D4T4 + D2T5  LD(AC) = D1T5 + D0T5 + D2T5  LD(E) = D1T5  LD(PC) = D4T4  Read = D1T4 + DOT5 + D2T4  X1 = D4T4  X7 = D1T4 + D0T4 + D2T4 |   For these CPU instructions : AND BUN BSA CMA  Translate their microoperations into signals  For each signal, derive a concise Boolean equation list alphabetically   |  |  | | --- | --- | |  |  | | Draw 3 Schematics according to the following Verilog modules   |  |  | | --- | --- | | Module Xmod(A, B)  Input [0:7] A;  Output[2:0] B;  Or(B[2], A[4], A[5], A[6], A[7]);  Or(B[1], A[2], A[3], A[6], A[7]);  Or(B[0], A[1], A[3], A[5], A[7]);  Endmodule  Module Ymod (B, C);  Input [2:0] B;  Output C;  Wire [0:2] and B  And(andB[0], B[2] , B[1]);  And(andB[1], B[2] , B[0]);  And(andB[2], B[1] , B[0]);  Or(C, and B[0], and B[1], andB[2]);  Endmodule  Module zmod(A,D);  Input[0:7] A;  Output[2:0]D;  Wire [2:0] B;  Wire C;  Xmod myX(A,B);  Ymod myY(B,C);  Xor(D[2], B[2], C);  Xor(D[1], B[1], C);  Xor(D[0], B[0], C);  Endmodule |  | |
| Given this gate-logic diagram, write Verilog code for this module. (No Test module needed. Arrays usage may be required.)  https://athena.ecs.csus.edu/~changw/137/midterm/XYmod.jpg  module Xmod(A, B, C);  input A, B;  output C;  wire invB;  not(invB, B);  and(C, A, invB);  endmodule  module Ymod(A, B, C, F);  input A, B, C;  output F;  wire invA, invB, X1out, X2out, andAB;  not(invA, A);  not(invB, B);  Xmod X1(C, invA, X1out);  Xmod X2(invB, A, X2out);  and(andAB, A, B);  or(F, X1out, X2out, andAB); | Given the Bus Diagram (Figure 5-4, p. 130) and the Control functions and Microoperations (Table 5-6, p. 159), derive the control signals needed to each of the CPU instructions. (Similar to exercise questions.)  selected instA: do-X, do-Y, ...  selected instB: do-X, do-Z, ...  selected instC: do-W, do-Y, ...  Derive timing for each of control signal derived from above. (Similar to exercise questions.) |