Nolan Anderson

CPE221-01

Dr. Rhonda Gaede

3/11/2020

Homework #5

**7.91**

**Cycle 1**: ER0\_C = 1

* Put R0 onto bus C

ER1\_D = 1,

* Put R1 onto bus D

**Cycle 2**: ALU(F3, F2, F1, F0) = 0,0,1,0, M\_ALU\_B, M\_R1, CR1

* Perform addition on R0 and R1 and put the value back into R1.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**7.92**

**Cycle 1:** EPC\_C = 1

* PC moves onto bus C

**Cycle 2:**  M\_ALU\_B;

* PC moves to bus B

**Cycle 3:** CMAR, Read = 1

* MAR gets PC and the memory is read.

**Cycle 4:**  M\_MBR, CIR

* The instruction register gets the memory of PC.

**Cycle 5:**  Incrementor, M\_PC

* Increment the value of PC.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**7.95**

**Cycle 1:**  EIR\_C = 1

* M moves to bus C

**Cycle 2:**  M\_ALU\_B

* M moves to bus B

**Cycle 3:** CMAR, Read = 1

* MAR gets M and reads the address of the data

**Cycle 4:**  M\_MBR, CMBR, EMBR\_C = 1

* M moves through the MBR and gets put onto bus C

**Cycle 5:** ER0\_D = 1

* R0 moves to bus D

**Cycle 6**: ALU(F3, F2, F1, F0) = 0,0,1,0, M\_ALU\_B, M\_R1, CR0

* Perform addition on M and R0 and put the value back into R0.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**7.96**

**Cycle 1:**  ER1\_C = 1

* R1 moves to bus C

**Cycle 2:**  M\_ALU\_B = 1

* R1 moves to bus B

**Cycle 3:**  CMAR, Read = 1

* MAR gets R1 and reads the address of the data

**Cycle 4:** M\_MBR, CMBR, EMBR\_C = 1

* R1 moves through the MBR and gets put onto bus C

**Cycle 5:** ER0\_D = 1

* R0 moves to bus D

**Cycle 6**: ALU(F3, F2, F1, F0) = 0,0,1,0, M\_ALU\_B, M\_R0, CR0

* Perform addition on M[R1] and R0 and put the value back into R0.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**7.97**

**Cycle 1:** ER0\_B = 1

* R0 moves to bus B

**Cycle 2:**  M\_ALU, CR1 = 1

- R0 moves to R1

**Cycle 3:**  EIR\_B = 1, M\_ALU

* M moves to bus A

**Cycle 4:**  CMAR = 1, Read = 1

* M moves to MAR

**Cycle 5:**  M\_MBR, CMBR, EMBR\_B = 1

* M[M] moves to bus B

**Cycle 6:**  M\_ALU, CMAR, Read = 1

* M[M] moves to MAR and the memory is read

**Cycle 7:** M\_MBR, CMBR, EMBR\_B = 1

- M[M[M]] moves to bus B

**Cycle 8:** ALU(F1,F0), 1, 0, M\_ALU, CR0

- Perform addition and move the data back into R0.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**7.98**

**Cycle 1:**  EIR\_B = 1

* M moves to bus B

**Cycle 2:** ALU(F1,F0), 1, 0, M\_ALU

* M+R1 moves to bus A

**Cycle 3:**  CMAR, Read =1

* Read the address of M+R1, result is M[M+R1]

**Cycle 4:**  M\_MBR, CMBR = 1

* M[M+R1] Moves to MBR

**Cycle 5:**  ER0\_B = 1

* R0 moves to bus B

**Cycle 6:**  M\_ALU, CR1 = 1

- R0 moves to R1

**Cycle 7:**  EMBR\_B = 1

* M[M+R1] moves to bus B

**Cycle 8:**  ALU(F1,F0), 1, 0, M\_ALU, CR0

- Perform addition on M[M+R1]+R0 and move the data back into R0.

**7.99**

|  |  |  |  |
| --- | --- | --- | --- |
| 3-Address | 2-Address | 1-Address | 0-Address |
| SUB A, C, D  MUL A, A, B  DIV A, A, E  ADD A, A, F  MUL A, A, G | LDR A, C  SUB A, D  MUL A, B  DIV. A, E  ADD. A, F  DIV. A, G | LDR C  SUB. D  MUL. B  DIV. E  ADD. F  MUL. G  STR. A | PUSH C  PUSH. D SUB.  PUSH. B MUL PUSH. E DIV PUSH. F ADD PUSH. G MUL POP. A |