Note: The numbers in this document are expressed in the hexadecimal or binary format.

**Content of microprogram ⎯ all fields for the program counter lecture**

|  |  |
| --- | --- |
| Note 1: | This week’s microinstruction design emphasizes the following bits:  **uP.1D,1C = sel\_opcode or operand,**  **uP 1B = flag value for disabling offset,**  **uP.1A,19,18 = sel\_PC in,**  **uP 17, 16, 15 = sel of flag** |
| Note 2: | The new instructions in the test programs for this week include：  **MOV direct,#imm**  **SJMP offset**  **LJMP addr16**  **JC offset**  **JNC offset**  You can try to design the microinstructions of these assembly instructions by yourself before looking at the teacher’s answer. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit no. (Hex) | 1F | 1E | 1D | 1C | 1B | 1A | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 0F | 0E | 0D | 0C | 0B | 0A | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |  |  |  |  |  |  |  |
| Bit no. (Decimal) | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| μPM  Addr |  |  | Sel\_ opcode or operand | | Flag value for disabling offset | sel PC in (= instru. Length) | | | Sel of flag | | | En of μPC counting | Sel\_ RAM addr | | Sel\_ RAM Din | |  | En\_ RAM Addr Lat | En\_ RAM Dout Lat | Str RAM |  |  | Sel\_ ALU Bin | | Sel\_ ALU | | | |  |  | En\_Cy update | En\_Acc update | μP code in hexa-decimal | Assembly  **(Green: correct)**  (Others: not yet) | Machine code | Length | Cycles  (Ours) | Cycles  (Official) |  |
| 00 |  |  | - | - | - | 0 | 0 | 1 | - | - | - | 0 | - | - | - | - | - | - | - | 0 | - | - | - | - | - | - | - | - |  |  | 0 | 0 | 01000000 | NOP | 00000000 | 1 | 1 | 1 |  |
| 01 |  |  | 0 | 1 | - | 0 | 1 | 0 | - | - | - | 0 | - | - | - | - |  | - | - | 0 |  |  | 0 | 1 | 0 | 0 | 1 | 1 |  |  | 0 | 1 | 12000131 | MOV A,#imm | 01110100 immediate | 2 | 1 | 1 | All the MOV instructions |
| 02 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MOV A,direct | 11100101 direct | 2 | 1 | 1 |
| 03 |  |  | - | - | - | 0 | 0 | 1 | - | - | - | 0 | 1 | 0 | - | - |  | 1 | - | 0 |  |  | 0 | 0 | 0 | 0 | 1 | 1 |  |  | 0 | 1 | 01084031 | MOV A,Rn | 11101nnn | 1 | 1 | 1 |
| 04 |  |  | - | - | - | 0 | 0 | 0 | - | - | - | 1 | 1 | 1 | - | - |  | 1 | 1 | 0 |  |  | - | - |  | - | - | - |  |  | 0 | 0 | 001C6000 | MOV A,@Ri | 1110011i | 1 | 2 | 1 |
| 05 |  |  | - | - | - | 0 | 0 | 1 | - | - | - | 0 | 0 | 0 | - | - |  | 1 | - | 0 |  |  | 0 | 0 | 0 | 0 | 1 | 1 |  |  | 0 | 1 | 01004031 |
| 06 |  |  | - | - | - | 0 | 0 | 1 | - | - | - | 0 | 1 | 0 | 0 | 1 |  | 1 | - | 1 |  |  | - | - |  | - | - | - |  |  | 0 | 0 | 01095000 | MOV Rn,A | 11111nnn | 1 | 1 | 1 |
| 07 |  |  | 0 | 1 | - | 0 | 1 | 0 | - | - | - | 0 | 1 | 0 | 1 | 0 | - | 1 | - | 1 | - | - | - | - | - | - | - | - |  |  | 0 | 0 | 120A5000 | MOV Rn,#imm | 01111nnn immediate | 2 |  | 1 |
| 08 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MOV Rn,direct | 10101nnn direct | 2 |  | 2 |
| 09 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0A |  |  | 0 | 1 | - | 0 | 1 | 0 | - | - | - | 0 | 0 | 1 | 0 | 1 | - | 1 | - | 1 |  |  | - | - |  | - | - | - |  |  | 0 | 0 | 12055000 | MOV direct,A | 11110101 direct | 2 | 1 | 1 |
| 0B |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MOV direct,#imm | 01110101 direct immediate | 3 | 2 | 2 |
| 0C |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0D |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MOV direct,Rn | 10001nnn direct | 2 |  | 2 |
| 0E |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0F |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MOV direct,direct | 10000101 src\_direct, dest\_direct | 3 | 2 | 2 |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MOV direct,@Ri |  | 2 |  | 2 |
| 12 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MOV @Ri,#imm | 0111011n immediate | 2 |  | 1 |
| 14 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 |  |  | - | - | - | 0 | 0 | 0 | - | - | - | 1 | 1 | 1 | - | - |  | 1 | 1 | 0 |  |  | - | - |  | - | - | - |  |  | 0 | 0 | 001C6000 | MOV @Ri,A | 1111011i | 1 | 2 | 1 |
| 16 |  |  | - | - | - | 0 | 0 | 1 | - | - | - | 0 | 0 | 0 | 0 | 1 |  | 1 | - | 1 |  |  | - | - |  | - | - | - |  |  | 0 | 0 | 01015000 |  |  |  |  |
| 17 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MOV @Ri,direct | 1010011i | 2 |  | 2 |
| 18 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 19 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MOV DPTR, #imm | 10010000 imm15-8 imm7-0 | 3 |  | 2 |
| 1A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1B |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MOV bit, C | 10010010 bit | 2 |  | 2 |
| 1C |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1D |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MOV C, bit | 10100010 bit | 2 |  | 1 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Bit no. (Hex) | 1F | 1E | 1D | 1C | 1B | 1A | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 0F | 0E | 0D | 0C | 0B | 0A | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |  |  |  |  |  |  |  |
|  |  |  | Sel\_ opcode or operand | | Flag value for disabling offset | sel PC in (= instru. Length) | | | Sel of flag | | | En of μPC counting | Sel\_ RAM addr | | Sel\_ RAM Din | |  | En\_ RAM Addr Lat | En\_ RAM Dout Lat | Str RAM |  |  | Sel\_ ALU Bin | | Sel\_ ALU | | | |  |  | En\_Cy update | En\_Acc update | μP code in hexa-decimal | Assembly | Machine code | Length | Cycles  (Ours) | Cycles  (Official) |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 24 |  |  | - | - | - | 0 | 0 | 1 | - | - | - | 0 | 1 | 0 | - | - | - | 1 | - | 0 | - |  |  |  |  |  |  | 0 | - | - | 1 | 1 | 01084003 | ADD A,Rn | 00101nnn | 1 | 1 | 1 |  |
| 25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ADD A,direct | 00100101 direct | 2 | 1 | 1 |  |
| 26 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | SUBB A,direct | 10010101 direct | 2 | 1 | 1 |  |
| 27 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 28 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | XRL A,#imm | 01100100 immediate | 2 |  | 1 |  |
| 29 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | XRL A,direct | 01100101 direct | 2 |  | 1 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 40 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | SJMP offset  (Official 8051 needs 2 cycle.) | 10000000 offset | 2 | 1 | 2 |  |
| 41 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 42 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | LJMP addr16 | 00000010 A15-A8, A7-A0 | 3 | 2 | 2 |  |
| 43 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 44 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | JC offset  (Official 8051 needs 2 cycle.) | 01000000 offset | 2 | 1 | 2 |  |
| 45 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 46 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | JNC offset  (Official 8051 needs 2 cycle.) | 01010000 offset | 2 | 1 | 2 |  |
| 47 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | JZ offset  (Official 8051 needs 2 cycle.) | 01100000 offset | 2 |  | 2 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | JNZ offset  (Official 8051 needs 2 cycle.) | 01110000 offset | 2 |  | 2 |  |
| Bit no. (Hex) | 1F | 1E | 1D | 1C | 1B | 1A | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 0F | 0E | 0D | 0C | 0B | 0A | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |  |  |  |  |  |  |  |
|  |  |  | Sel\_ opcode or operand | | Flag value for disabling offset | sel PC in (= instru. Length) | | | Sel of flag | | | En of μPC counting | Sel\_ RAM addr | | Sel\_ RAM Din | |  | En\_ RAM Addr Lat | En\_ RAM Dout Lat | Str RAM |  |  | Sel\_ ALU Bin | | Sel\_ ALU | | | |  |  | En of Cy update | En\_ Acc update | μP code in hexa-decimal | Assembly | Machine code | Length | Cycles  (Ours) | Cycles  (Official) |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |
| --- |
| Note:  -1. A 1-cycle instruction requires 1 μP word. A 2-cycle instruction requires 2 μP words.  -2. “En of μPC counting” = 0 for the ending μP word of any instruction. It = 1 for the non-ending μP words of any multi-cycle instruction.  -3. An all-zero μP word will terminate the program execution, because its “Sel of mux for PC-in selection” = 000, which means that the next instruction is the current instruction. The μP word of an NOP instruction is different to an all-zero μP word in that its “Sel of mux for PC-in selection” = 001, so the next instruction is the one following the NOP instruction.  -4. “SJMP offset” only requires 1 cycle in this design. It requires 2 cycles in the official 8051 CPU. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Offset control  ( For instruction with “uP. 1A, 19, 18, = sel PC in” = ’100’ or ‘101’) | | | | | | | | |
| uP. nn = Sel of flag | | 000 | 001 | 010 | 011 | 100 | 101 | Next address |
| Flag selected | | No flag selected | CY | Bit | Acc == 0 | X – Y == 0 | X – 1 == 0.  X = X – 1 |
| **Instruction** | **uP. mm, = Flag value for disabling offset** |
| SJMP | ⎯ | ⎯ |  |  |  |  |  | PC + 2 + offset |
| long jump | ⎯ | ⎯ |  |  |  |  |  | Long jump address |
| JC, | 0 |  | CY = 0 |  |  |  |  | PC + 2 |
| CY = 1 | PC + 2 + offset |
| JNC | 1 |  | CY = 0 |  |  |  |  | PC + 2 + offset |
| CY = 1 |  |  |  |  | PC + 2 |
| JB, | 0 |  |  | Bit = 0 |  |  |  | PC + 3 |
| Bit = 1 | PC + 3 + offset |
| JNB, | 1 |  |  | Bit = 0 |  |  |  | PC + 3 + offset |
| Bit = 1 | PC + 3 |
| JBC, | 0 |  |  | Bit = 0 |  |  |  | PC + 3 |
| Bit = 1 | PC + 3 + offset |
| ---------- |  |  |  |  |  |  |  |  |
| JZ, | 0 |  |  |  | 0 (Acc ≠ 0) |  |  | PC + 2 |
| 1 (Acc == 0) | PC + 2 + offset |
| JNZ | 1 |  |  |  | 0 (Acc ≠ 0) |  |  | PC + 2 + offset |
| 1 (Acc == 0) | PC + 2 |
| CJNE | 1 |  |  |  |  | 0 (x – y ≠ 0) |  | PC + 3 + offset |
| 1 ( X – Y == 0) | PC + 3 |
| DJNZ | 1 |  |  |  |  |  | 0 (x – 1 ≠ 0) | PC + 3 + offset |
| 1 (X – 1 == 0) | PC + 3 |