
UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Pertama
Sidang Akademik 2008/2009

November 2008

EEE320 – MIKROPEMROSES II

Masa: 2 jam

Sila pastikan bahawa kertas peperiksaan ini mengandungi TUJUH muka surat dan LIMA muka surat LAMPIRAN yang bercetak sebelum anda memulakan peperiksaan ini.

Kertas soalan ini mengandungi EMPAT soalan.

Jawab TIGA soalan.

Mulakan jawapan anda untuk setiap soalan pada muka surat yang baru.

Agihan markah bagi setiap soalan diberikan di sudut sebelah kanan soalan berkenaan.

Jawab semua soalan dalam bahasa Malaysia atau bahasa Inggeris atau kombinasi kedua-duanya.

1. (a) Apakah alamat bit ketiga pada alamat 'byte' 21H di dalam memori data dalaman 8051.

What is the bit address of bit-3 in byte address 21H in the 8051's internal data memory.

(10%)

- (b) Apakah alamat bit paling bererti pada alamat 'byte' 22H di dalam memori data dalaman 8051.

What is the bit address of the most significant bit at byte address 22H in the 8051's internal data memory.

(10%)

- (c) Apakah daftar bank yang aktif selepas pelaksanaan setiap arahan di bawah.

What is the active register bank after execution of each of the following instruction.

(i) MOV PSW, # 08H

(ii) MOV PSW, # 0FDH

(iii) MOV PSW, # 18H

(iv) MOV PSW, # 10H

(20%)

- (d) Sekiranya 8051 beroperasi dengan kristal 4MHz, berapakah masa untuk satu kitar mesin.

If an 8051 is operating with 4MHz crystal, what is the duration of a machine cycle.

(30%)

- (e) Kenal pasti posisi bit dan alamat 'byte' untuk setiap arahan SETB di bawah.

Identify the bit position and byte address for each of the following SETB instruction.

(i) SETB 37H

(ii) SETB 77H

(iii) SETB 0F7H

(iv) SETB 0A8H

(v) SETB 84H

(30%)

2. (a) Dapatkan bilangan pusingan berikut dilaksanakan.

Find the number of times the following loop is performed.

```
          MOV    R6, #200
          : MOV    R5, #100
BACK 1 : DJNZ   R5, BACK
BACK    DJNZ   R6, BACK 1
```

(10%)

- (b) Berdasarkan program bahasa perhimpunan di bawah, tentukan lokasi memori yang terlibat dan apakah kandungan terakhir lokasi tersebut.

For assembly language program given below, find the affected memory locations and what are the final contents of these affected locations.

```
          MOV    R0, #10
REP      : MOV    @R0, # 55H
          INC    R0
          CJNE  R0, # 20H, REP
LOOP    : MOV    @R0, # 0AAH
          DEC    R0
          CJNE  R0, # 5FH, LOOP
          END
```

(50%)

- (c) Tulis arahan bahasa perhimpunan untuk menambah dua nombor 16-bit. Andaikan nombor 16-bit tersebut disimpan di lokasi 30H : 31H dan 40H : 41H. Simpan nombor dari operasi tersebut ke lokasi 50H : 51H.

Write the assembly language instruction to add two 16-bit numbers. Assume the 16-bit numbers are stored in 30H : 31H and 40H : 41H. Store the sum of these number in 50H : 51H.

(40%)

3. (a) Tulis aturcara bahasa penghimpun untuk menghasilkan gelombang segiempat 500Hz daripada bit ke empat pada port 2 menggunakan pemasa 0. Isyarat yang dihasilkan mempunyai kitaran tugas 25%. Langkah yang disebabkan oleh arahan dalam gelung pengiraan mesti dimasukkan dalam pengiraan langkah masa untuk mendapatkan masa yang tepat bagi isyarat yang dihasilkan.

Write an assembly language program to generate a 500Hz square wave from bit 4 of port 2 using Timer 0. The generated signal should have a duty cycle of 25%. The delay due to instructions in the counting loop must be included in the calculation of time delay to get an accurate timing of the generated signal.

(50%)

- (b) Andaikan pin $\overline{\text{INT1}}$ disambungkan kepada suis dengan keadaan normal pada logik tinggi. Apabila ia berada pada logik rendah, LED akan dinyalakan. LED disambungkan kepada P1.3 dengan keadaan normalnya adalah terpadam. Apabila ia dinyalakan, ia akan menyala sehingga beberapa saat. Selagi suis ditekan kepada logik rendah, LED akan terus menyala. Lukis gambarajah blok dan tulis aturcara untuk melaksanakan tugas yang dinyatakan di atas.

Assume that the $\overline{\text{INT1}}$ pin is connected to a switch that is normally high. Whenever it goes low, it should turn on an LED. The LED is connected to P1.3 and is normally off. When it is turned on it should stay on for a fraction of a second. As long as the switch is pressed low, the LED should stay on. Draw a block diagram and write a program to perform the above mentioned task.

(50%)

4. (a) Rekabentuk sistem berasaskan mikropengawal 8051 dengan keperluan seperti berikut:

Design a 8051 microcontroller based system with the following requirement:

- (i) Dua peranti RAM (8K). Alamat untuk RAM pertama ialah 2000H hingga 2FFFH. Alamat untuk RAM kedua ialah 4000H hingga 4FFFH.

Two 8K RAM chips available. The address for the first RAM is 2000H to 2FFFH. The address for the second RAM is 4000H to 4FFFH.

- (ii) Terdapat dua peranti I/O. Alamat untuk peranti pertama ialah 6000H hingga 6FFFH dan alamat untuk peranti kedua ialah 7000H hingga 7FFFH.

Two I/O devices available. The address for the first device is 6000H to 6FFFH and the address for the second device is 7000H to 7FFFH.

(50%)

- (b) Tulis aturcara bahasa penghimpun untuk mikropengawal 8051, untuk menghantar huruf A secara serial pada kadar 'baud' 4800 secara berterusan dengan frekuensi kristal 11.0592 MHz.

Write an assembly language program for the 8051, to transmit the letter A in serial at 4800 baud rate, continuously with crystal frequency of 11.0592 MHz.

(50%)

oooo00oooo

PSW Register

| | | | | | | | |
|----|----|----|-----|-----|----|---|---|
| CY | AC | FO | RSI | RSO | OV | - | P |
|----|----|----|-----|-----|----|---|---|

MCS[®]-51 INSTRUCTION SET

8051 Instruction Set Summary

Interrupt Response Time: Refer to Hardware Description Chapter.

Instructions that Affect Flag Settings(1)

| Instruction | Flag | | | Instruction | Flag | | |
|-------------|------|----|----|-------------|------|----|----|
| | C | OV | AC | | C | OV | AC |
| ADD | X | X | X | CLR C | O | | |
| ADDC | X | X | X | CPL C | X | | |
| SUBB | X | X | X | ANL C,bit | X | | |
| MUL | O | X | | ANL C,/bit | X | | |
| DIV | O | X | | ORL C,bit | X | | |
| DA | X | | | ORL C,bit | X | | |
| RRC | X | | | MOV C,bit | X | | |
| RLC | X | | | CJNE | X | | |
| SETB C | 1 | | | | | | |

(1) Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Note on instruction set and addressing modes:

- Rn — Register R7–R0 of the currently selected Register Bank.
- direct — 8-bit internal data location's address. This could be an Internal Data RAM location (0–127) or a SFR [i.e., I/O port, control register, status register, etc. (128–255)].
- @Ri — 8-bit internal data RAM location (0–255) addressed indirectly through register R1 or R0.
- #data — 8-bit constant included in instruction.
- #data 16 — 16-bit constant included in instruction.
- addr 16 — 16-bit destination address. Used by LCALL & LJMP. A branch can be anywhere within the 64K-byte Program Memory address space.
- addr 11 — 11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.
- rel — Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is –128 to +127 bytes relative to first byte of the following instruction.
- bit — Direct Addressed bit in Internal Data RAM or Special Function Register.

| Mnemonic | Description | Byte | Oscillator Period |
|------------------------------|--|------|-------------------|
| ARITHMETIC OPERATIONS | | | |
| ADD A,Rn | Add register to Accumulator | 1 | 12 |
| ADD A,direct | Add direct byte to Accumulator | 2 | 12 |
| ADD A,@Ri | Add indirect RAM to Accumulator | 1 | 12 |
| ADD A,#data | Add immediate data to Accumulator | 2 | 12 |
| ADDC A,Rn | Add register to Accumulator with Carry | 1 | 12 |
| ADDC A,direct | Add direct byte to Accumulator with Carry | 2 | 12 |
| ADDC A,@Ri | Add indirect RAM to Accumulator with Carry | 1 | 12 |
| ADDC A,#data | Add immediate data to Acc with Carry | 2 | 12 |
| SUBB A,Rn | Subtract Register from Acc with borrow | 1 | 12 |
| SUBB A,direct | Subtract direct byte from Acc with borrow | 2 | 12 |
| SUBB A,@Ri | Subtract indirect RAM from ACC with borrow | 1 | 12 |
| SUBB A,#data | Subtract immediate data from Acc with borrow | 2 | 12 |
| INC A | Increment Accumulator | 1 | 12 |
| INC Rn | Increment register | 1 | 12 |
| INC direct | Increment direct byte | 2 | 12 |
| INC @Ri | Increment direct RAM | 1 | 12 |
| DEC A | Decrement Accumulator | 1 | 12 |
| DEC Rn | Decrement Register | 1 | 12 |
| DEC direct | Decrement direct byte | 2 | 12 |
| DEC @Ri | Decrement indirect RAM | 1 | 12 |

All mnemonics copyrighted ©Intel Corporation 1980

8051 Instruction Set Summary (Continued)

| Mnemonic | Description | Byte | Oscillator Period |
|--|--|------|-------------------|
| ARITHMETIC OPERATIONS (Continued) | | | |
| INC DPTR | Increment Data Pointer | 1 | 24 |
| MUL AB | Multiply A & B | 1 | 48 |
| DIV AB | Divide A by B | 1 | 48 |
| DA A | Decimal Adjust Accumulator | 1 | 12 |
| LOGICAL OPERATIONS | | | |
| ANL A,Rn | AND Register to Accumulator | 1 | 12 |
| ANL A,direct | AND direct byte to Accumulator | 2 | 12 |
| ANL A,@Ri | AND indirect RAM to Accumulator | 1 | 12 |
| ANL A,#data | AND immediate data to Accumulator | 2 | 12 |
| ANL direct,A | AND Accumulator to direct byte | 2 | 12 |
| ANL direct,#data | AND immediate data to direct byte | 3 | 24 |
| ORL A,Rn | OR register to Accumulator | 1 | 12 |
| ORL A,direct | OR direct byte to Accumulator | 2 | 12 |
| ORL A,@Ri | OR indirect RAM to Accumulator | 1 | 12 |
| ORL A,#data | OR immediate data to Accumulator | 2 | 12 |
| ORL direct,A | OR Accumulator to direct byte | 2 | 12 |
| ORL direct,#data | OR immediate data to direct byte | 3 | 24 |
| ^RL A,Rn | Exclusive-OR register to Accumulator | 1 | 12 |
| XRL A,direct | Exclusive-OR direct byte to Accumulator | 2 | 12 |
| XRL A,@Ri | Exclusive-OR indirect RAM to Accumulator | 1 | 12 |
| XRL A,#data | Exclusive-OR immediate data to Accumulator | 2 | 12 |
| XRL direct,A | Exclusive-OR Accumulator to direct byte | 2 | 12 |
| XRL direct,#data | Exclusive-OR immediate data to direct byte | 3 | 24 |
| CLR A | Clear Accumulator | 1 | 12 |
| CPL A | Complement Accumulator | 1 | 12 |

| Mnemonic | Description | Byte | Oscillator Period |
|---------------------------------------|--|------|-------------------|
| LOGICAL OPERATIONS (Continued) | | | |
| RL A | Rotate Accumulator Left | 1 | 12 |
| RLC A | Rotate Accumulator Left through the Carry | 1 | 12 |
| RR A | Rotate Accumulator Right | 1 | 12 |
| RRC A | Rotate Accumulator Right through the Carry | 1 | 12 |
| SWAP A | Swap nibbles within the Accumulator | 1 | 12 |
| DATA TRANSFER | | | |
| MOV A,Rn | Move register to Accumulator | 1 | 12 |
| MOV A,direct | Move direct byte to Accumulator | 2 | 12 |
| MOV A,@Ri | Move indirect RAM to Accumulator | 1 | 12 |
| MOV A,#data | Move immediate data to Accumulator | 2 | 12 |
| MOV Rn,A | Move Accumulator to register | 1 | 12 |
| MOV Rn,direct | Move direct byte to register | 2 | 24 |
| MOV Rn,#data | Move immediate data to register | 2 | 12 |
| MOV direct,A | Move Accumulator to direct byte | 2 | 12 |
| MOV direct,Rn | Move register to direct byte | 2 | 24 |
| MOV direct,direct | Move direct byte to direct byte | 3 | 24 |
| MOV direct,@Ri | Move indirect RAM to direct byte | 2 | 24 |
| MOV direct,#data | Move immediate data to direct byte | 3 | 24 |
| MOV @Ri,A | Move Accumulator to indirect RAM | 1 | 12 |

All mnemonics copyrighted © Intel Corporation 1980

8051 Instruction Set Summary (Continued)

| Mnemonic | Description | Byte | Oscillator Period |
|----------------------------------|--|------|-------------------|
| DATA TRANSFER (Continued) | | | |
| MOV @Ri,direct | Move direct byte to indirect RAM | 2 | 24 |
| MOV @Ri,#data | Move immediate data to indirect RAM | 2 | 12 |
| MOV DPTR,#data16 | Load Data Pointer with a 16-bit constant | 3 | 24 |
| MOVC A,@A+DPTR | Move Code byte relative to DPTR to Acc | 1 | 24 |
| MOVC A,@A+PC | Move Code byte relative to PC to Acc | 1 | 24 |
| MOVX A,@Ri | Move External RAM (8-bit addr) to Acc | 1 | 24 |
| MOVX A,@DPTR | Move External RAM (16-bit addr) to Acc | 1 | 24 |
| MOVX @Ri,A | Move Acc to External RAM (8-bit addr) | 1 | 24 |
| MOVX @DPTR,A | Move Acc to External RAM (16-bit addr) | 1 | 24 |
| PUSH direct | Push direct byte onto stack | 2 | 24 |
| POP direct | Pop direct byte from stack | 2 | 24 |
| XCH A,Rn | Exchange register with Accumulator | 1 | 12 |
| XCH A,direct | Exchange direct byte with Accumulator | 2 | 12 |
| XCH A,@Ri | Exchange indirect RAM with Accumulator | 1 | 12 |
| XCHD A,@Ri | Exchange low-order Digit indirect RAM with Acc | 1 | 12 |

| Mnemonic | Description | Byte | Oscillator Period |
|--------------------------------------|---------------------------------------|------|-------------------|
| BOOLEAN VARIABLE MANIPULATION | | | |
| CLR C | Clear Carry | 1 | 12 |
| CLR bit | Clear direct bit | 2 | 12 |
| SETB C | Set Carry | 1 | 12 |
| SETB bit | Set direct bit | 2 | 12 |
| CPL C | Complement Carry | 1 | 12 |
| CPL bit | Complement direct bit | 2 | 12 |
| ANL C,bit | AND direct bit to CARRY | 2 | 24 |
| ANL C,/bit | AND complement of direct bit to Carry | 2 | 24 |
| ORL C,bit | OR direct bit to Carry | 2 | 24 |
| ORL C,/bit | OR complement of direct bit to Carry | 2 | 24 |
| MOV C,bit | Move direct bit to Carry | 2 | 12 |
| MOV bit,C | Move Carry to direct bit | 2 | 24 |
| JC rel | Jump if Carry is set | 2 | 24 |
| JNC rel | Jump if Carry not set | 2 | 24 |
| JB bit,rel | Jump if direct Bit is set | 3 | 24 |
| JNB bit,rel | Jump if direct Bit is Not set | 3 | 24 |
| JBC bit,rel | Jump if direct Bit is set & clear bit | 3 | 24 |
| PROGRAM BRANCHING | | | |
| ACALL addr11 | Absolute Subroutine Call | 2 | 24 |
| LCALL addr16 | Long Subroutine Call | 3 | 24 |
| RET | Return from Subroutine | 1 | 24 |
| RETI | Return from interrupt | 1 | 24 |
| AJMP addr11 | Absolute Jump | 2 | 24 |
| LJMP addr16 | Long Jump | 3 | 24 |
| SJMP rel | Short Jump (relative addr) | 2 | 24 |

All mnemonics copyrighted ©Intel Corporation 1980

8051 Instruction Set Summary (Continued)

| Mnemonic | Description | Byte | Oscillator Period |
|--------------------------------------|--|------|-------------------|
| PROGRAM BRANCHING (Continued) | | | |
| JMP @A + DPTR | Jump indirect relative to the DPTR | 1 | 24 |
| JZ rel | Jump if Accumulator is Zero | 2 | 24 |
| JNZ rel | Jump if Accumulator is Not Zero | 2 | 24 |
| CJNE A,direct,rel | Compare direct byte to Acc and Jump if Not Equal | 3 | 24 |
| CJNE A,#data,rel | Compare immediate to Acc and Jump if Not Equal | 3 | 24 |

| Mnemonic | Description | Byte | Oscillator Period |
|--------------------------------------|---|------|-------------------|
| PROGRAM BRANCHING (Continued) | | | |
| CJNE Rn,#data,rel | Compare immediate to register and Jump if Not Equal | 3 | 24 |
| CJNE @Ri,#data,rel | Compare immediate to indirect and Jump if Not Equal | 3 | 24 |
| DJNZ Rn,rel | Decrement register and Jump if Not Zero | 2 | 24 |
| DJNZ direct,rel | Decrement direct byte and Jump if Not Zero | 3 | 24 |
| NOP | No Operation | 1 | 12 |

All mnemonics copyrighted © Intel Corporation 1980