**NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY**
**FACULTY OF APPLIED SCIENCE**
**COMPUTER SCIENCE DEPARTMENT**
**MAY EXAMINATIONS 2002**

**SUBJECT:** MICROPROCESSORS AND EMBEDDED SYSTEMS
**CODE:** SCS2202

INSTRUCTION TO CANDIDATES
The Question Paper consists of seven (7) questions
Answer any five questions.

**Time: 3 hours**

## QUESTION ONE

a) Using the block diagram, describe the features of an MC 6800 microprocessor. [10]

b) Discuss briefly, giving examples, the instruction formats used with MC6800 microprocessor. [10]

## QUESTION TWO

a) Discuss the various addressing modes used in an MC6800 microprocessor. [6]

b) Identify the addressing modes used in each of the following MC6800 instructions. [6]

(i) CLRA
(ii) LDX%11100001
(iii) ADDA $02
(iv) EORB #$0A

c) Give the instruction format in hexadecimal for each of the above instructions in 4(b) and show the organisation in memory.

## QUESTION THREE

Write an assembly language program for an MC 6800 to multiply two numbers less than 16. The two numbers are located in $70 and $71 and the result should be stored in $72. [20]

**QUESTION FOUR**

a) With the aid of a diagram, describe the operation of a peripheral interface adapter. [10]

b) List the steps needed to initialise a port. [5]

c) Describe the steps taken to clear and set bits in a control register. [5]

**QUESTION FIVE**

a) What are the roles of the stack pointer and the program counter? [4]

b) What are the three phases of executing instructions within the MC6800? [6]

c) Discuss the actions taken by an MC6800 microprocessor upon receiving an interrupt. [10]

**QUESTION SIX**

For the MC6800 assembly language program given below:

a) Find the task performed by the program
b) Show the organisation of the program in the memory assuming that the program is stored in the memory in locations starting in address 2020#.
c) Find the memory size required to store the program.

```
        LDX #08H
        LDAB #01H
        STX 1000H
LOOP    INCB
        STAB 1001H
        E0RB 1000H
        BNE  LOOP
        DEC 1001H
        LDAA 1001
        DECA
        STAA 80H
        SWI
```

**QUESTION SEVEN**

Write a simple assembly language program that inputs 100 bytes from a port with data register address 1AH and control register address 1BH and stores them in memory locations starting from 2A00H if the data is positive and in memory locations starting from 3A00H if the data is negative. [20]

**END OF QUESTION PAPER**

GOOD LUCK!

# Appendix A

## Instruction Set for 6800 Microprocessor

**Accumulator and Memory**

| Op code | Action | Immed Hex | # | Direct Hex | # | Extend Hex | # | Index Hex | # | Inher Hex | # |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDA | Add | 8B | 2 | 9B | 3 | BB | 4 | AB | 5 | | |
| ADDB | | CB | 2 | DB | 3 | FB | 4 | EB | 5 | | |
| ABA | Add Acmltrs | | | | | | | | | 1B | 2 |
| ADCA | Add wth carry | 89 | 2 | 99 | 3 | B9 | 4 | A9 | 5 | | |
| ADCB | | C9 | 2 | D9 | 3 | F9 | 4 | E9 | 5 | | |
| ANDA | And | 84 | 2 | 94 | 3 | B4 | 4 | A4 | 5 | | |
| ANDB | | C4 | 2 | D4 | 3 | F4 | 4 | E4 | 5 | | |
| BITA | Bit Test | 85 | 2 | 95 | 3 | B5 | 4 | A5 | 5 | | |
| BITB | | C5 | 2 | D5 | 3 | F5 | 4 | E5 | 5 | | |
| CLR | Clear | | | | | 7F | 6 | 6F | 7 | | |
| CLRA | | | | | | | | | | 4F | 2 |
| CLRB | | | | | | | | | | 5F | 2 |
| CMPA | Compare | 81 | 2 | 91 | 3 | B1 | 4 | A1 | 5 | | |
| CMPB | | C1 | 2 | D1 | 3 | F1 | 4 | E1 | 5 | | |
| CBA | Compare Acmltrs | | | | | | | | | 11 | 2 |
| COM | Complement, 1's | | | | | 73 | 6 | 63 | 7 | | |
| COMA | | | | | | | | | | 43 | 2 |
| COMB | | | | | | | | | | 53 | 2 |
| NEG | Complement, 2's | | | | | 70 | | 60 | | | |
| NEGA | (Negate) | | | | | | | | | 40 | 2 |
| NEGB | | | | | | | | | | 50 | 2 |
| DAA | Decimal Adjust,A | | | | | | | | | 19 | 2 |
| DEC | Decrement | | | | | 7A | 6 | 6A | 7 | | |
| DECA | | | | | | | | | | 4A | 2 |
| DECB | | | | | | | | | | 5A | 2 |
| EORA | Exclusive OR | 88 | 2 | 98 | 3 | B8 | 4 | A8 | 5 | | |
| EORB | | C8 | 2 | D8 | 3 | F8 | 4 | E8 | 5 | | |
| INC | Increment | | | | | 7C | 6 | 6C | 7 | | |
| INCA | | | | | | | | | | 4C | 2 |
| INCB | | | | | | | | | | 5C | 2 |
| LDAA | Load Acmltr | 86 | 2 | 96 | 3 | B6 | 4 | A6 | 5 | | |

Index Register and Stack

| Op code | Action | Immed Hex | # | Direct Hex | # | Extend Hex | # | Index Hex | # | Inher Hex | # |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CPX | Compare Index Reg | 8C | 3 | 9C | 4 | BC | 5 | AC | 6 | | |
| DEX | Decrement Ind Reg | | | | | | | | | 09 | 4 |
| DES | Decrement SP | | | | | | | | | 34 | 4 |
| INX | Increment SP | | | | | | | | | 08 | 4 |
| LDX | Load Index Reg | CE | 3 | DE | 4 | FE | 5 | EE | 6 | | |
| LDS | Load Stack Pntr | 8E | 3 | 9E | 4 | BE | 5 | AE | 6 | | |
| STX | Store Index Reg | | | DF | 5 | FF | 6 | EF | 7 | | |
| STS | Store Stack Pntr | | | 9F | 5 | BF | 6 | AF | 7 | | |
| TXS | Indx Reg to SP | | | | | | | | | 35 | 4 |
| TSX | SP to Ind Reg | | | | | | | | | 30 | 4 |

Jump and Branch

| Op code | Action | Relative Hex | # | Extend Hex | # | Index Hex | # | | Inher Hex | # |
|---|---|---|---|---|---|---|---|---|---|---|
| BRA | Branch always | | | | | | | | 20 | 4 |
| BCC | Branch if carry clear | | | | | | | | 24 | 4 |
| BCS | Branch if carry set | | | | | | | | 25 | 4 |
| BEQ | Branch if = Zero | | | | | | | | 27 | 4 |
| BGE | Branch if >= Zero | | | | | | | | 2C | 4 |
| BGT | Branch if > Zero | | | | | | | | 2E | 4 |
| BHI | Branch if Higher | | | | | | | | 22 | 4 |
| BLE | Branch if <= Zero | | | | | | | | 2F | 4 |
| BLS | If Lower or same | | | | | | | | 23 | 4 |
| BLT | Branch if < Zero | | | | | | | | 2D | 4 |
| BMI | Branch if Minus | | | | | | | | 2B | 4 |
| BNE | Branch if not = Zero | | | | | | | | 26 | 4 |
| BVC | If Overflow Clear | | | | | | | | 28 | 4 |
| BVS | If Overflow Set | | | | | | | | 29 | 4 |
| BPL | Branch if Plus | | | | | | | | 2A | 4 |
| BSR | Br to Subroutine | | | | | | | | 8D | 8 |
| JMP | Jump | | | 7E | 3 | 6E | 4 | | | |
| JSR | Jump to subroutine | | | BD | 3 | AD | 4 | | | |

| Op code | Action | Immed Hex | # | Direct Hex | # | Extend Hex | # | Index Hex | # | Inher Hex | # |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LDAB | | C6 | 2 | D6 | 3 | F6 | 4 | E6 | 6 | | |
| ORAA | Or, inclusive | 8A | 2 | 9A | 3 | BA | 4 | AA | 5 | | |
| ORAB | | CA | 2 | DA | 3 | FA | 4 | EA | 5 | | |
| PSHA | Push Data | | | | | | | | | 36 | 4 |
| PSHB | | | | | | | | | | 37 | 4 |
| PULA | Pull Data | | | | | | | | | 32 | 4 |
| PULB | | | | | | | | | | 33 | 4 |
| ROL | Rotate Left | | | | | 79 | 6 | 69 | 7 | | |
| ROLA | | | | | | | | | | 49 | 2 |
| ROLB | | | | | | | | | | 59 | 2 |
| ROR | Rotate Right | | | | | 76 | 6 | 66 | 7 | | |
| RORA | | | | | | | | | | 46 | 2 |
| RORB | | | | | | | | | | 56 | 2 |
| ASL | Shift Left, | | | | | 78 | 6 | 68 | 7 | | |
| ASLA | Arithmetic | | | | | | | | | 48 | 2 |
| ASLB | | | | | | | | | | 58 | 2 |
| ASR | Shift right, | | | | | 77 | 6 | 67 | 7 | | |
| ASRA | Arithmetic | | | | | | | | | 47 | 2 |
| ASRB | | | | | | | | | | 57 | 2 |
| LSR | Shift Right, | | | | | 74 | 6 | 64 | 7 | | |
| LSRA | Logic | | | | | | | | | 44 | 2 |
| LSRB | | | | | | | | | | 54 | 2 |
| STAA | Store Acmltr. | | | 97 | 4 | B7 | 5 | A7 | 6 | | |
| STAB | | | | D7 | 4 | F7 | 5 | E7 | 6 | | |
| SUBA | Subtract | 80 | 2 | 90 | 3 | B0 | 4 | A0 | 5 | | |
| SUBB | | C0 | 2 | D0 | 3 | F0 | 4 | E0 | 5 | | |
| SBA | | | | | | | | | | 10 | 2 |
| SBCA | Subtract with carry | 82 | 2 | 92 | 3 | B2 | 4 | A2 | 5 | | |
| SBCB | | C2 | 2 | D2 | 3 | F2 | 4 | E2 | 5 | | |
| TAB | Transfer Acmltrs | | | | | | | | | 16 | 2 |
| TBA | | | | | | | | | | 17 | 2 |
| TST | Test, Zero or Minus | | | | | 7D | 6 | 6D | 7 | | |
| TSTA | | | | | | | | | | 4D | 2 |
| TSTB | | | | | | | | | | 5D | 2 |

| Op code | Action | Relative Hex # | Extend Hex # | Index Hex # | | Inher Hex # |
|---|---|---|---|---|---|---|
| NOP | No Operation | | | | | 01 2 |
| RTI | Return From Int | | | | | 3B 10 |
| RTS | Return From Subr | | | | | 39 5 |
| SWI | Software Interrupt | | | | | 3F 12 |
| WAI | Wait for Interrupt | | | | | 3E 9 |

**Conditions Code Register**

| Ops | Action | | Inher Hex # |
|---|---|---|---|
| CLC | Clear Carry | | 0C 2 |
| CLI | Clear interrupt | | 0E 2 |
| CLV | Clear Overflow | | 0A 2 |
| SEC | Set Carry | | 0D 2 |
| SEI | Set Interrupt Mask | | 0F 2 |
| SEC | Set Overflow | | 0B 2 |
| TAP | Acmltr to A CCR | | 06 2 |
| TPA | CCR to Acmltr A | | 07 2 |