Document No : UniKL MFI_SD_AC41
Revision No: 02
Effective Date: 01 December 2008

CONFIDENTIAL

SET A

UNIVERSITI KUALA LUMPUR
Malaysia France Institute

FINAL EXAMINATION
JANUARY 2010 SESSION

SUBJECT CODE	:	FSD 23102
SUBJECT TITLE	:	MICROPROCESSOR
LEVEL	:	DIPLOMA
TIME / DURATION	:	4.00pm – 6.00pm (2 HOURS)
DATE	:	04 MAY 2010

INSTRUCTIONS TO CANDIDATES

1. Please read the instructions given in the question paper CAREFULLY.
2. This question paper is printed on both sides of the paper.
3. Please write your answers on the answer booklet provided.
4. Answer should be written in blue or black ink except for sketching, graphic and illustration.
5. This questions paper consists of TWO (2) sections. Section A and B. Answer ALL questions in section A. For sections B, answer TWO (2) questions only.
6. Answer all questions in English.
7. *ALL types of calculators are STRICTLY PROHIBITED during the examination.*

THERE ARE 8 PAGES OF QUESTIONS AND 4 PAGES OF APPENDICES, EXCLUDING THIS PAGE.

SECTION A (Total: 60 marks)

INSTRUCTION: Answer ALL questions.
Please use the answer booklet provided.

Question 1

(a) What is the main function of Microprocessor?

(2 marks)

(b) Give four main elements in Microprocessor Unit.

(4 marks)

(c) There are two main applications of Microprocessor such as Embedded System and Reprogrammable System. Explain the meaning of Embedded System and give two examples of application.

(3 marks)

(d) Explain the meaning of DMA Control in MC6809 and list two approaches in this operation.

(4 marks)

(e) What is the meaning of DAA instruction in assembly language?

(2 marks)

(f) List three 2 bytes registers in microprocessor 6809.

(3 marks)

(g) Write the instruction in assembly language that will load the Index Register, X with a value $ABCD.

(1 mark)

(h) By referring to Question 1(g) above, what is the type of addressing mode for that instruction?

(1 mark)

(i) Assembly Language or High Level Language are known as _____ while Machine Language is known as _____.

(2 marks)

(j) What is the function of memory and explain ROM and RAM?

(3 marks)

Question 2

Convert and perform the arithmetic operation below. You are required to show the conversion procedure algorithmically.

(a) Convert **$25D** to decimal form.

(3 marks)

(b) Convert %**1101111110** to hexadecimal form.

(3 marks)

(c) By using *two's complement*, perform the following operation.

Note: *Your calculations should be in 8-bit format for integer numbers.*

$2F - &19

(6 marks)

(d) By referring to Question 2(c) above, what happens to the status of N-bit and H-bit in CCR register and explain why?

(4 marks)

Question 3

Explain the function of each component below:

(a) Instruction Register (IR).

(2 marks)

(b) Condition Code Register (CCR).

(2 marks)

Question 4

Identify the types of addressing mode of each instruction below:

(a) BRA !3

(1 mark)

(b) DEX

(1 mark)

(c) LDX #$FFFF

(1 mark)

(d) ADDA $6789

(1 mark)

(e) STAA $01, X

(1 mark)

Question 5

Figure 1 shows the Bus Contention in Microprocessor 6809.

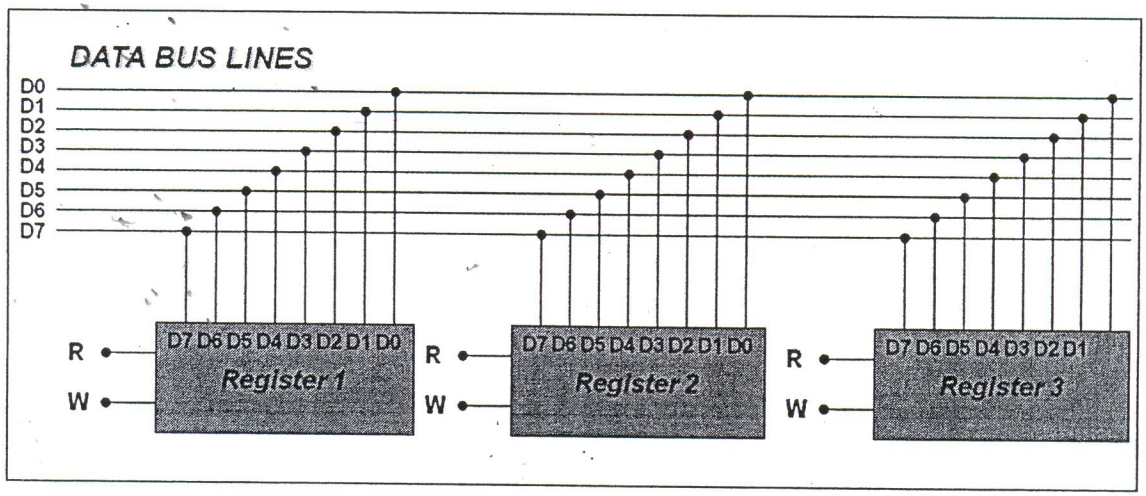


Figure 1: Illustration of Bus Contention

(a) Describe briefly on Bus Contention.

(3 marks)

(b) Determine the solution to overcome this Bus Contention.

(4 marks)

(c) What is the meaning of Three State Output?

(3 marks)

SECTION B (Total: 40 marks)

INSTRUCTION: Answer TWO (2) questions only.
Please use the answer booklet provided.

Question 6

(a) There are three types of operations that are executed by microprocessor such as read, write and internal operations. Explain briefly the meaning of write operation in microprocessor.

(2 marks)

(b) Draw the timing diagram during write operation.

(8 marks)

(c) By referring to Question 6(b) above, describe the process of write operation in microprocessor.

(10 marks)

Question 7

(a) Design a flow chart to search for negative numbers in memory locations starting from $0301 to $03FF and the result of count is saved in $0500.

(4 marks)

(b) Write an assembly language code, complete with comments to represent your algorithm written in Question 7(a). The program also must be started at location $0200.

Note: *Refer to Instruction Set in Appendices.*

(10 marks)

(c) List two flags that will be affected in CCR and explain your statement?

(6 marks)

Question 8

(a) Design a flow chart to save ascending order numbers from $00 to $AA into successive memory locations starting at $0400 to $04AA. The first value $00 is saved in location $0400 and the last value, $AA is saved in location $04AA.

(4 marks)

(b) Write an assembly language code, complete with comments to represent your algorithm written in Question 8(a). The program also must be started at location $0200.

Note: *Refer to Instruction Set in Appendices.*

(10 marks)

(c) Explain the function of BNE instruction in Microprocessor 6809 and give an example.

(6 marks)

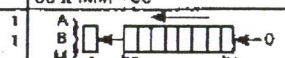
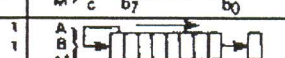
END OF QUESTION

APPENDICES

INSTRUCTION SET

MC6809

PROGRAMMING AID

Instruction	Forms	Addressing Modes: Immediate Op	Immediate ~	Immediate #	Direct Op	Direct ~	Direct #	Indexed Op	Indexed ~	Indexed #	Extended Op	Extended ~	Extended #	Inherent Op	Inherent ~	Inherent #	Description	5 H	3 N	2 Z	1 V	0 C
ABX														3A	3	1	B + X → X (Unsigned)	•	•	•	•	•
ADC	ADCA	89	2	2	99	4	2	A9	4+	2+	B9	5	3				A + M + C → A	↕	↕	↕	↕	↕
	ADCB	C9	2	2	D9	4	2	E9	4+	2+	F9	5	3				B + M + C → B	↕	↕	↕	↕	↕
ADD	ADDA	8B	2	2	9B	4	2	AB	4+	2+	BB	5	3				A + M → A	↕	↕	↕	↕	↕
	ADDB	CB	2	2	DB	4	2	EB	4+	2+	FB	5	3				B + M → B	↕	↕	↕	↕	↕
	ADDD	C3	4	3	D3	6	2	E3	6+	2+	F3	7	3				D + M:M + 1 → D	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	4	2	A4	4+	2+	B4	5	3				A ∧ M → A	•	↕	↕	0	•
	ANDB	C4	2	2	D4	4	2	E4	4+	2+	F4	5	3				B ∧ M → B	•	↕	↕	0	•
	ANDCC	1C	3	2													CC ∧ IMM → CC					7
ASL	ASLA													48	2	1		↕	↕	↕	↕	↕
	ASLB													58	2	1		↕	↕	↕	↕	↕
	ASL				08	6	2	68	6+	2+	78	7	3					↕	↕	↕	↕	↕
ASR	ASRA													47	2	1		↕	↕	↕	•	↕
	ASRB													57	2	1		↕	↕	↕	•	↕
	ASR				07	6	2	67	6+	2+	77	7	3					↕	↕	↕	•	↕
BIT	BITA	85	2	2	95	4	2	A5	4+	2+	B5	5	3				Bit Test A (M ∧ A)	•	↕	↕	0	•
	BITB	C5	2	2	D5	4	2	E5	4+	2+	F5	5	3				Bit Test B (M ∧ B)	•	↕	↕	0	•
CLR	CLRA													4F	2	1	0 → A	•	0	1	0	0
	CLRB													5F	2	1	0 → B	•	0	1	0	0
	CLR				0F	6	2	6F	6+	2+	7F	7	3				0 → M	•	0	1	0	0
CMP	CMPA	81	2	2	91	4	2	A1	4+	2+	B1	5	3				Compare M from A	↕	↕	↕	↕	↕
	CMPB	C1	2	2	D1	4	2	E1	4+	2+	F1	5	3				Compare M from B	↕	↕	↕	↕	↕
	CMPD	10 83	5	4	10 93	7	3	10 A3	7+	3+	10 B3	8	4				Compare M:M + 1 from D	•	↕	↕	↕	↕
	CMPS	11 8C	5	4	11 9C	7	3	11 AC	7+	3+	11 BC	8	4				Compare M:M + 1 from S	•	↕	↕	↕	↕
	CMPU	11 83	5	4	11 93	7	3	11 A3	7+	3+	11 B3	8	4				Compare M:M + 1 from U	•	↕	↕	↕	↕
	CMPX	8C	4	3	9C	6	2	AC	6+	2+	BC	7	3				Compare M:M + 1 from X	•	↕	↕	↕	↕
	CMPY	10 8C	5	4	10 9C	7	3	10 AC	7+	3+	10 BC	8	4				Compare M:M + 1 from Y	•	↕	↕	↕	↕
COM	COMA													43	2	1	$\overline{A} \rightarrow A$	•	↕	↕	0	1
	COMB													53	2	1	$\overline{B} \rightarrow B$	•	↕	↕	0	1
	COM				03	6	2	63	6+	2+	73	7	3				$\overline{M} \rightarrow M$	•	↕	↕	0	1
CWAI		3C	≥20	2													CC ∧ IMM → CC Wait for Interrupt					7
DAA														19	2	1	Decimal Adjust A	•	↕	↕	0	↕
DEC	DECA													4A	2	1	A − 1 → A	•	↕	↕	↕	•
	DECB													5A	2	1	B − 1 → B	•	↕	↕	↕	•
	DEC				0A	6	2	6A	6+	2+	7A	7	3				M − 1 → M	•	↕	↕	↕	•
EOR	EORA	88	2	2	98	4	2	A8	4+	2+	B8	5	3				A ⊻ M → A	•	↕	↕	0	•
	EORB	C8	2	2	D8	4	2	E8	4+	2+	F8	5	3				B ⊻ M → B	•	↕	↕	0	•
EXG	R1, R2	1E	8	2													R1 ↔ R2[2]	•	•	•	•	•
INC	INCA													4C	2	1	A + 1 → A	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	↕	↕	↕	•
	INC				0C	6	2	6C	6+	2+	7C	7	3				M + 1 → M	•	↕	↕	↕	•
JMP					0E	3	2	6E	3+	2+	7E	4	3				EA[3] → PC	•	•	•	•	•
JSR					9D	7	2	AD	7+	2+	BD	8	3				Jump to Subroutine	•	•	•	•	•
LD	LDA	86	2	2	96	4	2	A6	4+	2+	B6	5	3				M → A	•	↕	↕	0	•
	LDB	C6	2	2	D6	4	2	E6	4+	2+	F6	5	3				M → B	•	↕	↕	0	•
	LDD	CC	3	3	DC	5	2	EC	5+	2+	FC	6	3				M:M + 1 → D	•	↕	↕	0	•
	LDS	10 CE	4	4	10 DE	6	3	10 EE	6+	3+	10 FE	7	4				M:M + 1 → S	•	↕	↕	0	•
	LDU	CE	3	3	DE	5	2	EE	5+	2+	FE	6	3				M:M + 1 → U	•	↕	↕	0	•
	LDX	8E	3	3	9E	5	2	AE	5+	2+	BE	6	3				M:M + 1 → X	•	↕	↕	0	•
	LDY	10 8E	4	4	10 9E	6	3	10 AE	6+	3+	10 BE	7	4				M:M + 1 → Y	•	↕	↕	0	•
LEA	LEAS							32	4+	2+							EA[3] → S	•	•	•	•	•
	LEAU							33	4+	2+							EA[3] → U	•	•	•	•	•
	LEAX							30	4+	2+							EA[3] → X	•	•	↕	•	•
	LEAY							31	4+	2+							EA[3] → Y	•	•	↕	•	•

LEGEND:

OP Operation Code (Hexadecimal)
~ Number of MPU Cycles
# Number of Program Bytes
+ Arithmetic Plus
− Arithmetic Minus
• Multiply
$\overline{M}$ Complement of M
→ Transfer Into
H Half-carry (from bit 3)
N Negative (sign bit)
Z Zero result
V Overflow, 2's complement
C Carry from ALU
↕ Test and set if true, cleared otherwise
• Not Affected
CC Condition Code Register
: Concatenation
∨ Logical or
∧ Logical and
⊻ Logical Exclusive or

INSTRUCTION SET
MC6809

PROGRAMMING AID (CONTINUED)

Instruction	Forms	Addressing Modes: Immediate Op	~	#	Direct Op	~	#	Indexed[1] Op	~	#	Extended Op	~	#	Inherent Op	~	#	Description	5 H	3 N	2 Z	1 V	0 C
LSL	LSLA													48	2	1	A, B, M: C b7 b0 ← 0	•	↕	↕	↕	↕
	LSLB													58	2	1		•	↕	↕	↕	↕
	LSL				08	6	2	68	6+	2+	78	7	3					•	↕	↕	↕	↕
LSR	LSRA													44	2	1	A, B, M: 0 → b7 b0 C	•	0	↕	•	↕
	LSRB													54	2	1		•	0	↕	•	↕
	LSR				04	6	2	64	6+	2+	74	7	3					•	0	↕	•	↕
MUL														3D	11	1	A × B → D (Unsigned)	•	•	↕	•	9
NEG	NEGA													40	2	1	$\overline{A}+1 \rightarrow A$	↕	↕	↕	↕	↕
	NEGB													50	2	1	$\overline{B}+1 \rightarrow B$	↕	↕	↕	↕	↕
	NEG				00	6	2	60	6+	2+	70	7	3				$\overline{M}+1 \rightarrow M$	↕	↕	↕	↕	↕
NOP														12	2	1	No Operation	•	•	•	•	•
OR	ORA	8A	2	2	9A	4	2	AA	4+	2+	BA	5	3				A ∨ M → A	•	↕	↕	0	•
	ORB	CA	2	2	DA	4	2	EA	4+	2+	FA	5	3				B ∨ M → B	•	↕	↕	0	•
	ORCC	1A	3	2													CC ∨ IMM → CC	↕	↕	↕	↕	↕
PSH	PSHS	34	5+[4]	2													Push Registers on S Stack	•	•	•	•	•
	PSHU	36	5+[4]	2													Push Registers on U Stack	•	•	•	•	•
PUL	PULS	35	5+[4]	2													Pull Registers from S Stack	•	•	•	•	•
	PULU	37	5+[4]	2													Pull Registers from U Stack	•	•	•	•	•
ROL	ROLA													49	2	1	A, B, M: C b7 b0	•	↕	↕	↕	↕
	ROLB													59	2	1		•	↕	↕	↕	↕
	ROL				09	6	2	69	6+	2+	79	7	3					•	↕	↕	↕	↕
ROR	RORA													46	2	1	A, B, M: C b7 b0	•	↕	↕	•	↕
	RORB													56	2	1		•	↕	↕	•	↕
	ROR				06	6	2	66	6+	2+	76	7	3					•	↕	↕	•	↕
RTI														3B	6-15	1	Return From Interrupt					
RTS														39	5	1	Return from Subroutine	•	•	•	•	•
SBC	SBCA	82	2	2	92	4	2	A2	4+	2+	B2	5	3				A − M − C → A	↕	↕	↕	↕	↕
	SBCB	C2	2	2	D2	4	2	E2	4+	2+	F2	5	3				B − M − C → B	↕	↕	↕	↕	↕
SEX														1D	2	1	Sign Extend B into A	•	↕	↕	↕	•
ST	STA				97	4	2	A7	4+	2+	B7	5	3				A → M	•	↕	↕	0	•
	STB				D7	4	2	E7	4+	2+	F7	5	3				B → M	•	↕	↕	0	•
	STD				DD	5	2	ED	5+	2+	FD	6	3				D → M M+1	•	↕	↕	0	•
	STS				10	6	3	10	6+	3+	10	7	4				S → M M+1	•	↕	↕	0	•
					DF			EF			FF											
	STU				DF	5	2	EF	5+	2+	FF	6	3				U → M M+1	•	↕	↕	0	•
	STX				9F	5	2	AF	5+	2+	BF	6	3				X → M M+1	•	↕	↕	0	•
	STY				10	6	3	10			10	7	4				Y → M M+1	•	↕	↕	0	•
					9F			AF	6+	3+	BF											
SUB	SUBA	80	2	2	90	4	2	A0	4+	2+	B0	5	3				A − M → A	↕	↕	↕	↕	↕
	SUBB	C0	2	2	D0	4	2	E0	4+	2+	F0	5	3				B − M → B	↕	↕	↕	↕	↕
	SUBD	83	4	3	93	6	2	A3	6+	2+	B3	7	3				D − M M+1 → D	•	↕	↕	↕	↕
SWI	SWI[6]													3F	19	1	Software Interrupt 1	•	•	•	•	•
	SWI2[6]													10 3F	20	2	Software Interrupt 2	•	•	•	•	•
	SWI3[6]													11 3F	20	1	Software Interrupt 3	•	•	•	•	•
SYNC														13	≥4	1	Synchronize to Interrupt	•	•	•	•	•
TFR	R1, R2	1F	6	2													R1 → R2[2]	•	•	•	•	•
TST	TSTA													4D	2	1	Test A	•	↕	↕	0	•
	TSTB													5D	2	1	Test B	•	↕	↕	0	•
	TST				0D	6	2	6D	6+	2+	7D	7	3				Test M	•	↕	↕	0	•

NOTES:

1. This column gives a base cycle and byte count. To obtain total count, add the values obtained from the INDEXED ADDRESSING MODE table, Table 2.
2. R1 and R2 may be any pair of 8 bit or any pair of 16 bit registers.
   The 8 bit registers are: A, B, CC, DP
   The 16 bit registers are: X, Y, U, S, D, PC
3. EA is the effective address.
4. The PSH and PUL instructions require 5 cycles plus 1 cycle for each byte pushed or pulled.
5. 5(6) means: 5 cycles if branch not taken, 6 cycles if taken (Branch instructions).
6. SWI sets I and F bits. SWI2 and SWI3 do not affect I and F.
7. Conditions Codes set as a direct result of the instruction.
8. Value of half-carry flag is undefined.
9. Special Case – Carry set if b7 is SET.

INSTRUCTION SET
MC6809

PROGRAMMING AID (CONTINUED)

Branch Instructions

Instruction	Forms	Addressing Mode Relative OP	~	#	Description	5 H	3 N	2 Z	1 V	0 C
BCC	BCC	24	3	2	Branch C = 0	•	•	•	•	•
	LBCC	10 24	5(6)	4	Long Branch C = 0	•	•	•	•	•
BCS	BCS	25	3	2	Branch C = 1	•	•	•	•	•
	LBCS	10 25	5(6)	4	Long Branch C = 1	•	•	•	•	•
BEQ	BEQ	27	3	2	Branch Z = 1	•	•	•	•	•
	LBEQ	10 27	5(6)	4	Long Branch Z = 0	•	•	•	•	•
BGE	BGE	2C	3	2	Branch ≥ Zero	•	•	•	•	•
	LBGE	10 2C	5(6)	4	Long Branch ≥ Zero	•	•	•	•	•
BGT	BGT	2E	3	2	Branch > Zero	•	•	•	•	•
	LBGT	10 2E	5(6)	4	Long Branch > Zero	•	•	•	•	•
BHI	BHI	22	3	2	Branch Higher	•	•	•	•	•
	LBHI	10 22	5(6)	4	Long Branch Higher	•	•	•	•	•
BHS	BHS	24	3	2	Branch Higher or Same	•	•	•	•	•
	LBHS	10 24	5(6)	4	Long Branch Higher or Same	•	•	•	•	•
BLE	BLE	2F	3	2	Branch ≤ Zero	•	•	•	•	•
	LBLE	10 2F	5(6)	4	Long Branch ≤ Zero	•	•	•	•	•
BLO	BLO	25	3	2	Branch lower	•	•	•	•	•
	LBLO	10 25	5(6)	4	Long Branch Lower	•	•	•	•	•

Instruction	Forms	Addressing Mode Relative OP	~	#	Description	5 H	3 N	2 Z	1 V	0 C
BLS	BLS	23	3	2	Branch Lower or Same	•	•	•	•	•
	LBLS	10 23	5(6)	4	Long Branch Lower or Same	•	•	•	•	•
BLT	BLT	2D	3	2	Branch < Zero	•	•	•	•	•
	LBLT	10 2D	5(6)	4	Long Branch < Zero	•	•	•	•	•
BMI	BMI	2B	3	2	Branch Minus	•	•	•	•	•
	LBMI	10 2B	5(6)	4	Long Branch Minus	•	•	•	•	•
BNE	BNE	26	3	2	Branch Z = 0	•	•	•	•	•
	LBNE	10 26	5(6)	4	Long Branch Z = 0	•	•	•	•	•
BPL	BPL	2A	3	2	Branch Plus	•	•	•	•	•
	LBPL	10 2A	5(6)	4	Long Branch Plus	•	•	•	•	•
BRA	BRA	20	3	2	Branch Always	•	•	•	•	•
	LBRA	16	5	3	Long Branch Always	•	•	•	•	•
BRN	BRN	21	3	2	Branch Never	•	•	•	•	•
	LBRN	10 21	5	4	Long Branch Never	•	•	•	•	•
BSR	BSR	8D	7	2	Branch to Subroutine	•	•	•	•	•
	LBSR	17	9	3	Long Branch to Subroutine	•	•	•	•	•
BVC	BVC	28	3	2	Branch V = 0	•	•	•	•	•
	LBVC	10 28	5(6)	4	Long Branch V = 0	•	•	•	•	•
BVS	BVS	29	3	2	Branch V = 1	•	•	•	•	•
	LBVS	10 29	5(6)	4	Long Branch V = 1	•	•	•	•	•

SIMPLE BRANCHES

	OP	~	#
BRA	20	3	2
LBRA	16	5	3
BRN	21	3	2
LBRN	1021	5	4
BSR	8D	7	2
LBSR	17	9	3

SIMPLE CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
N = 1	BMI	2B	BPL	2A
Z = 1	BEQ	27	BNE	26
V = 1	BVS	29	BVC	28
C = 1	BCS	25	BCC	24

SIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r > m	BGT	2E	BLE	2F
r ≥ m	BGE	2C	BLT	2D
r = m	BEQ	27	BNE	26
r ≤ m	BLE	2F	BGT	2E
r < m	BLT	2D	BGE	2C

UNSIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r > m	BHI	22	BLS	23
r ≥ m	BHS	24	BLO	25
r = m	BEQ	27	BNE	26
r ≤ m	BLS	23	BHI	22
r < m	BLO	25	BHS	24

NOTES:
1. All conditional branches have both short and long variations.
2. All short branches are two bytes and require three cycles.
3. All conditional long branches are formed by prefixing the short branch opcode with $10 and using a 16-bit destination offset.
4. All conditional long branches require four bytes and six cycles if the branch is taken or five cycles if the branch is not taken.