INTI
INTERNATIONAL COLLEGE PENANG (507232-U)
LAUREATE INTERNATIONAL UNIVERSITIES

FINAL
Examination Paper

(COVER PAGE)

Session : JAN 2012

Programme : DIPLOMA IN ELECTRICAL AND ELECTRONIC ENGINEERING

Course : **EEE 2105: INTRODUCTION TO MICROPROCESSORS**

Date of Examination : 6 March 2012

Time : 11a.m. – 1p.m. Reading Time : Nil

Duration : 2 Hours

Special Instructions :

This paper consists of **SIX (6)** questions. Answer any **FOUR (4)** questions in the answer booklet provided. All questions carry equal marks.

Students are not allowed to remove the question papers from the examination venue.

Materials permitted :

NON-PROGRAMMABLE SCIENTIFIC CALCULATOR

Materials provided :

APPENDIX A (8086 Instruction Set Summary), APPENDIX B (ASCII Table), APPENDIX C (8255 PPI) and APPENDIX D (8253/8254 PIT)

Examiner(s) : **STEVEN KHOO**

Moderator : **CHAN TSE WEI**

*This paper consists of **6** printed pages, including the cover page.*

INTI INTERNATIONAL COLLEGE PENANG

DIPLOMA IN ELECTRICAL AND ELECTRONIC ENGINEERING PROGRAMME (DEE/I)

EEE 2105: INTRODUCTION TO MICROPROCESSORS
FINAL EXAMINATION: JAN2012 SESSION

Instructions: This paper consists of **SIX (6)** questions. Answer any **FOUR (4)** questions in the answer booklet provided. All questions carry equal marks.

Question 1

(a) Perform the following number system transformation. Show all workings clearly.

(i) $CA2.9E_H$ to octal

(3 marks)

(ii) 43976.3046875_{10} to hexadecimal

(5 marks)

(b) Give ONLY one alternative instruction for each of the following instructions:

(i) XOR AX, AX

(2 marks)

(ii) INC CX

(2 marks)

(iii) MOV BL, 2
MUL BL

(2 marks)

(iv) MOV BL, CL
MOV CL, AL
MOV AL, BL

(2 marks)

(c) Find the organization and bit capacity of the following memory chips.

(i) 13 address pins, 4 data pins SRAM

(3 marks)

(ii) 9 address pins, CAS, RAS, 4 data pins DRAM

(3 marks)

(d) Comment on the validity of the following instructions. Explain if it is not valid.

(i) MOV AX, AX

(ii) MOV BX, [AX]

(iii) MOV AH, [DI] + 3H

(3 marks)

Question 2

(a) Explain the difference between RCL and ROL instructions. Illustrate how a microprocessor rotates twice the data 5AH using these types of instruction with the aid of diagrams. Assume that the data is in register AL and carry flag (CF) is set initially.

(8 marks)

(b) At a certain moment the state of an 8086 microprocessor based system is as follows: (All values are in Hexadecimal)

Registers

	H	L
AX	2C	10
BX	00	14
CX	10	01
DX	2B	33
CS	01	00
DS	80	00
SS	40	10
ES	61	00
BP	00	1F
SP	00	08
IP	11	0A
SI	21	34
DI	FF	FF

Memory

8000B	12
8000A	34
80009	56
80008	78
80007	9A
80006	BC
80005	DE
80004	F0
80003	21
80002	43
80001	09
80000	67
7FFFF	54
7FFFE	DF
7FFFD	3C

4010E	6F
4010D	2D
4010C	C1
4010B	25
4010A	3F
40109	12
40108	FF
40107	FC
40106	E1
40105	10
40104	00
40103	36
40102	21
40101	88
40000	99

(i) What is the physical address of the next instruction to be executed?

(2 marks)

(ii) What is the physical address of the top of the stack?

(2 marks)

(iii) Draw a memory map for the 8086 microprocessor, indicating clearly the start and end addresses of the segments.

(4 marks)

(iv) Provide the registers and memory locations that are affected, and their new values after the execution of ADD [BX – 0AH], AX, also state the number of bytes used for this instruction.

(4 marks)

(c) Assume that AX = 7FFFH and BX = 0001H. Will the conditional jump to label "SUM" take place when the following commands execute? Justify your answer.

```
CMP   AX, BX
JG    SUM
```

(5 marks)

Question 3

(a) Calculate the time delay taken for the following instruction with an 8086 microprocessor running at 10MHz. Also calculate the number of bytes used to store this program.

```
        MOV  BL, 04H
        MOV  CL, BL
LOOP1:  DEC  CL
        JNZ  LOOP1
        JMP  LOOP2
        ADD  BL, BL
LOOP2:  HLT
```

(8 marks)

[Refer to APPENDIX A for the cycle time]

(b) Differentiate the following arithmetic instructions with the aid of calculation. Assuming AX = 53FCH and BX = 7A86H. Show all workings clearly including the result.

(i) DIV BL

(4 marks)

(ii) MUL BH

(4 marks)

(iii) IMUL BH

(4 marks)

Operation (byte): AL = AX ÷ operand, AH = remainder
Operation (word): AX = DX:AX ÷ operand, DX = remainder

Operation (byte): AX = AL × operand
Operation (word): DX:AX = AX × operand

(c) Draw a block diagram showing the interconnections of microcomputers' buses.

(5 marks)

Question 4

(a) A transmission system uses asynchronous serial data communication with 2 stop bits, and even parity. Draw the frame for extended ASCII character "L", "W", "N" using continuous character transmission, LSB being transmitted first. Calculate the total time it takes to transfer 900 characters if 1200bps is being used.

(8 marks)

[Refer to Appendix B for ASCII Codes]

(b) The 8255 PPI is configured as shown in Figure 4(b).

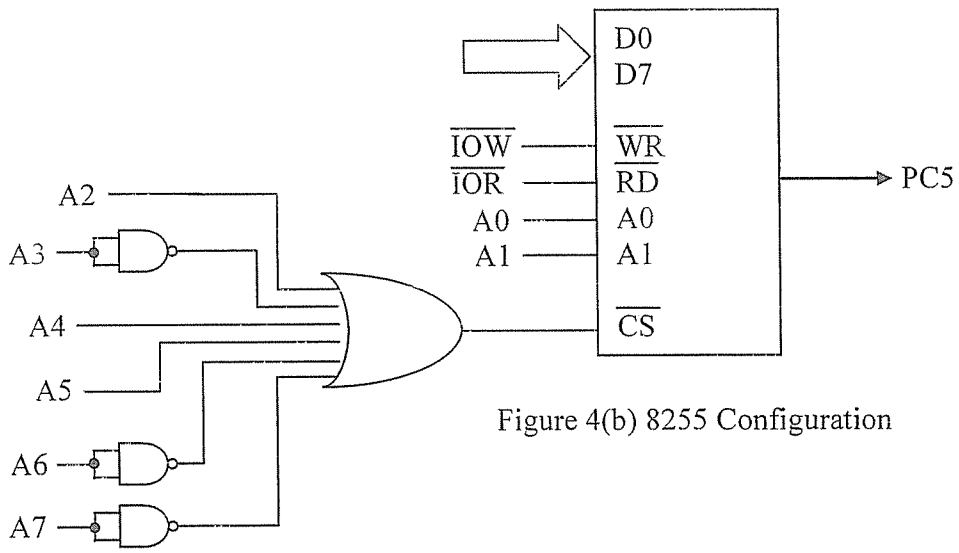


Figure 4(b) 8255 Configuration

(i) Find the port addresses and control register. Thus, program the PPI to set PC3 to high. Also include comments for any instruction used.

(4 marks)

(ii) Write a program to generate a square wave of 80% duty cycle by using bit 5 port C. Assume that the DELAY subroutine program is available. Also include comments for any instruction used.

(5 marks)

[Refer to Appendix C for 8255 PPI Control Word].

(c) Assume SP = FF2EH, AX = 1239H, BX = F53CH, CX = 123FH
With the aid of a memory map, show the content of stack and stack pointer after the execution of the following sequence of instructions:

```
PUSH AX
PUSH BX
PUSH CX
```

(6 marks)

(d) A 16-bit microprocessor has bit, byte or word processing capability. Identify the following codes in terms of the type of word.

(i) 96AA4611BE4C72FF
(ii) 3A567B41

(2 marks)

Question 5

(a) With the aid of a suitable diagram, illustrate step-by-step how DMA can be used to speed up CPU operation during the transfer of data from memory to an I/O device.

(11 marks)

(b) The bus cycle of the 8086/88 microprocessors consists of at least four clock periods, namely T_1, T_2, T_3 and T_4. Discuss in details what is happening in each T-state for a read bus cycle of memory in 8088.

(9 marks)

(c) Pin $\overline{CS}$ of a given 8253/8254 is activated by binary address A7 – A2 = 100101. Find the port addresses assigned to this 8253/8254 and the configuration for this PIT if the control register is programmed as follows:

```
MOV  AL, 00110110
OUT  97H, AL
```

(5 marks)

[Refer to Appendix D for 8253/8254 PIT Control Word].

Question 6

A certain microprocessor allocates addresses 1000H to 1FFFH for RAM 1, 3000H to 5FFFH for RAM 2, 7000H to 9FFFH for I/O and C000H to FFFFH for ROM.

(a) Draw the memory map for this microprocessor.

(1 mark)

(b) Determine the total ROM & RAM capacity.

(3 marks)

(c) How many different input/output devices can the assigned I/O capacity area accommodate if an I/O device uses 3 KB of memory space?

(2 marks)

(d) How many different input/output devices can this microchip efficiently accommodate if an I/O device uses 2 KB of memory space?

(4 marks)

(e) Using combination of logic gates and decoders, design the partial address decoding circuitry for the above memory and I/O devices. All design steps must be shown.

(15 marks)

– THE END –

EEE2105(F)/Jan12/Steven Khoo/27/12/11

APPENDICES BOOKLET

Instructions:
Students are NOT allowed to remove the appendices booklet from the examination venue and also NOT allowed to write anything on this booklet.

A: 8086 INSTRUCTION SET SUMMARY [8 pages]

B: ASCII CODES TABLE [1 page]

C: 8255 PPI (PROGRAMMABLE PERIPHERAL INTERFACE) [1 page]

D: 8253/8254 PIT (PROGRAMMABLE INTERVAL TIMER) [1 page]

APPENDIX A: 8086 INSTRUCTION SET SUMMARY

Instruction Set Summary

Mnemonic	Description	Clock cycles	Number of bytes	Flags O D I T S Z A P C	Page ref.
AAA	ASCII adjust for addition	4	1	u - - - u u x u x	76
AAD	ASCII adjust for division	60	2	u - - - x x u x u	76
AAM	ASCII adjust for multiplication	83	2	u - - - x x u x u	76
AAS	ASCII adjust for subtraction	4	1	u - - - u u x u x	76
ADC	Add with carry			x - - - x x x x x	65
	Register to register	3	2		
	Memory to register	9+EA	2–4		
	Register to memory	16+EA	2–4		
	Immediate to register	4	3–4		
	Immediate to memory	17+EA	3–6		
	Immediate to accumulator	4	2–3		
ADD	Addition			x - - - x x x x x	65
	Register to register	3	2		
	Memory to register	9+EA	2–4		
	Register to memory	16+EA	2–4		
	Immediate to register	4	3–4		
	Immediate to memory	17+EA	3–6		
	Immediate to accumulator	4	2–3		
AND	Logical AND			0 - - - x x u x 0	93
	Register to register	3	2		
	Memory to register	9+EA	2–4		
	Register to memory	16+EA	2–4		
	Immediate to register	4	3–4		
	Immediate to memory	17+EA	3–6		
	Immediate to accumulator	4	2–3		
CALL	Call a procedure			- - - - - - - - - -	157
	Intrasegment direct	19	3		
	Intrasegment indirect through register	16	2		
	Intrasegment indirect through memory	21+EA	2–4		
	Intersegment direct	28	5		
	Intersegment indirect	37+EA	2–4		
CBW	Convert byte to word	2	1	- - - - - - - - - -	68
CLC	Clear carry flag	2	1	- - - - - - - - - 0	92
CLD	Clear direction flag	2	1	- 0 - - - - - - -	92
CLI	Clear interrupt flag	2	1	- - 0 - - - - - -	92
CMC	Complement carry flag	2	1	- - - - - - - - - x	92
CMP	Compare			x - - - x x x x x	69
	Register to register	3	2		
	Memory to register	9+EA	2–4		
	Register to memory	9+EA	2–4		
	Immediate to register	4	3–4		
	Immediate to memory	10+EA	3–6		

Mnemonic	Description	Clock cycles	Number of bytes	Flags O D I T S Z A P C	Page ref.
	Immediate to accumulator	4	2–3		
CMPS/	Compare string/		1	x - - - x x x x x	208
CMPSB/	Compare byte string/				
CMPSW	Compare word string				
	Not repeated	22			
	Repeated	9 + 22/rep			
CWD	Convert word to double word	5	1	- - - - - - - - - -	68
DAA	Decimal adjust for addition	4	1	u - - - x x x x x	74
DAS	Decimal adjust for subtraction	4	1	u - - - x x x x x	74
DEC	Decrement by 1			x - - - x x x x -	69
	16-bit register	2	1		
	8-bit register	3	2		
	Memory	15 + EA	2–4		
DIV	Unsigned division			u - - - u u u u u	70
	8-bit register	80–90	2		
	16-bit register	144–162	2		
	8-bit memory	(86–96) + EA	2–4		
	16-bit memory	(150–168) + EA	2–4		
ESC	Escape			- - - - - - - - - -	457
	Register	2	2		
	Memory	8 + EA	2–4		
HLT	Halt	2	1	- - - - - - - - - -	91
IDIV	Integer division			u - - - u u u u u	70
	8-bit register	101–112	2		
	16-bit register	165–184	2		
	8-bit memory	(107–118) + EA	2–4		
	16-bit memory	(171–190) + EA	2–4		
IMUL	Integer multiplication			x - - - u u u u x	70
	8-bit register	80–98	2		
	16-bit register	128–154	2		
	8-bit memory	(86–104) + EA	2–4		
	16-bit memory	(134–160) + EA	2–4		
IN	Input from I/O port			- - - - - - - - - -	232
	Fixed port	10	2		
	Variable port	8	1		
INC	Increment by 1			x - - - x x x x -	69
	16-bit register	2	1		
	8-bit register	3	2		
	Memory	15 + EA	2–4		
INT	Interrupt			- - 0 0 - - - - -	172

Mnemonic	Description	Clock cycles	Number of bytes	Flags O D I T S Z A P C	Page ref.
	Type = 3	52	1		
	Type ≠ 3	51	2		
INTO	Interrupt if overflow		1	- - 0 0 - - - - -	172
	Interrupt is taken	53			
	Interrupt is not taken	4			
IRET	Return from interrupt	24	1	r r r r r r r r r	172
JA/	Jump if above/	16/4	2	- - - - - - - - -	80
JNBE	Jump if not below or equal				
JAE/	Jump if above or equal/	16/4	2	- - - - - - - - -	80
JNB/	Jump if not below/				
JNC	Jump if not carry				
JB/	Jump if below/	16/4	2	- - - - - - - - -	80
JNAE/	Jump if not above or equal/				
JC	jump if carry				
JBE/	Jump if below or equal/	16/4	2	- - - - - - - - -	80
JNA	Jump if not above				
JCXZ	Jump if CX is zero	18/6	2	- - - - - - - - -	88
JE/	Jump if equal/	16/4	2	- - - - - - - - -	80
JZ	Jump if zero				
JG/	Jump if greater/	16/4	2	- - - - - - - - -	80
JNLE	Jump if not less or equal				
JGE/	Jump if greater or equal/	16/4	2	- - - - - - - - -	80
JNL	Jump if not less				
JL/	Jump if less/	16/4	2	- - - - - - - - -	80
JNGE	Jump if not greater or equal				
JLE/	Jump if less or equal/	16/4	2	- - - - - - - - -	80
JNG	Jump if not greater				
JMP	Jump			- - - - - - - - -	84
	Intrasegment direct short	15	2		
	Intrasegment direct	15	3		
	Intersegment direct	15	5		
	Intrasegment indirect through memory	18+EA	2–4		
	Intrasegment indirect through register	11	2		
	Intersegment indirect	24+EA	2–4		
JNE/	Jump if not equal/	16/4	2	- - - - - - - - -	80
JNZ	Jump if not zero				
JNO	Jump if not overflow	16/4	2	- - - - - - - - -	80
JNP/	Jump if not parity/	16/4	2	- - - - - - - - -	80
JPO	Jump if parity odd				
JNS	Jump if not sign	16/4	2	- - - - - - - - -	80
JO	Jump if overflow	16/4	2	- - - - - - - - -	80
JP/	Jump if parity/	16/4	2	- - - - - - - - -	80
JPE	Jump if parity even				
JS	Jump if sign	16/4	2	- - - - - - - - -	80
LAHF	Load AH from flags	4	1	- - - - - - - - -	92
LDS	Load pointer using DS	16+EA	2–4	- - - - - - - - -	60
LEA	Load effective address	2+EA	2–4	- - - - - - - - -	60

Mnemonic	Description	Clock cycles	Number of bytes	Flags O D I T S Z A P C	Page ref.
LES	Load pointer using ES	16+EA	2–4	- - - - - - - - - -	60
LOCK	Lock bus	2	1	- - - - - - - - - -	455
LODS/	Load string/		1	- - - - - - - - - -	208
LODSB/	Load byte string/				
LODSW	Load word string				
	Not repeated	12			
	Repeated	9+13/rep			
LOOP	Loop	17/5	2	- - - - - - - - - -	88
LOOPE/	Loop if equal/	18/6	2	- - - - - - - - - -	88
LOOPZ	Loop if zero				
LOOPNE/	Loop if not equal/	19/5	2	- - - - - - - - - -	88
LOOPNZ	Loop if not zero				
MOV	Move			- - - - - - - - - -	60
	Accumulator to memory	10	3		
	Memory to accumulator	10	3		
	Register to register	2	2		
	Memory to register	8+EA	2–4		
	Register to memory	9+EA	2–4		
	Immediate to register	4	2–3		
	Immediate to memory	10+EA	3–6		
	Register to SS, DS, or ES	2	2		
	Memory to SS, DS, or ES	8+EA	2–4		
	Segment register to register	2	2		
	Segment register to memory	9+EA	2–4		
MOVS/	Move string/		1	- - - - - - - - - -	208
MOVSB/	Move byte string/				
MOVSW	Move word string				
	Not repeated	18			
	Repeated	9+17/rep			
MUL	Unsigned multiplication			x - - - u u u u x	70
	8-bit register	70–77	2		
	16-bit register	118–133	2		
	8-bit memory	(76–83) +EA	2–4		
	16-bit memory	(124–139) +EA	2–4		
NEG	Negate			x - - - x x x x x	69
	Register	3	2		
	Memory	16+EA	2–4		
NOP	No operation	3	1	- - - - - - - - - -	91
NOT	Logical NOT			- - - - - - - - - -	93
	Register	3	2		
	Memory	16+EA	2–4		
OR	Logical OR			0 - - - x x u x 0	93
	Register to register	3	2		
	Memory to register	9+EA	2–4		
	Register to memory	16+EA	2–4		

Mnemonic	Description	Clock cycles	Number of bytes	Flags O D I T S Z A P C	Page ref.
	Immediate to accumulator	4	2–3		
	Immediate to register	4	3–4		
	Immediate to memory	17+EA	3–6		
OUT	Output to I/O port			- - - - - - - - - -	232
	Fixed port	10	2		
	Variable port	8	1		
POP	Pop word off stack			- - - - - - - - - -	153
	Register	8	1		
	Segment register SS, DS, or ES	8	1		
	Memory	17+EA	2–4		
POPF	Pop flags off stack	8	1	r r r r r r r r r	153
PUSH	Push word onto stack			- - - - - - - - - -	153
	Register	11	1		
	Segment register	10	1		
	Memory	16+EA	2–4		
PUSHF	Push flags onto stack	10	1	- - - - - - - - - -	153
RCL	Rotate left through carry			x - - - - - - - x	98
	Register with single-shift	2	2		
	Register with variable-shift	8+4/bit	2		
	Memory with single-shift	15+EA	2–4		
	Memory with variable-shift	20+EA +4/bit	2–4		
RCR	Rotate right through carry			x - - - - - - - x	98
	Register with single-shift	2	2		
	Register with variable-shift	8+4/bit	2		
	Memory with single-shift	15+EA	2–4		
	Memory with variable-shift	20+EA +4/bit	2–4		
REP	Repeat string operation	2	1	- - - - - - - - - -	213
REPE/	Repeat operation while equal/	2	1	- - - - - - - - - -	213
REPZ	Repeat operation while zero				
REPNE/	Repeat operation while not equal/	2	1	- - - - - - - - - -	213
REPNZ	Repeat operation while not zero				
RET	Return from procedure			- - - - - - - - - -	157
	Intrasegment	8	1		
	Intrasegment with constant	12	3		
	Intersegment	18	1		
	Intersegment with constant	17	3		
ROL	Rotate left			x - - - - - - - x	98
	Register with single-shift	2	2		

Mnemonic	Description	Clock cycles	Number of bytes	Flags O D I T S Z A P C	Page ref.
	Register with variable-shift	8+4/bit	2		
	Memory with single-shift	15+EA	2–4		
	Memory with variable-shift	20+EA +4/bit	2–4		
ROR	Rotate right			x - - - - - - - x	98
	Register with single-shift	2	2		
	Register with variable-shift	8+4/bit	2		
	Memory with single-shift	15+EA	2–4		
	Memory with variable-shift	20+EA +4/bit	2–4		
SAHF	Store AH into flags	4	1	- - - - r r r r r	92
SAL/	Shift arithmetic left/			x - - - x x u x x	98
SHL	Shift logical left				
	Register with single-shift	2	2		
	Register with variable-shift	8+4/bit	2		
	Memory with single-shift	15+EA	2–4		
	Memory with variable-shift	20+EA +4/bit	2–4		
SAR	Shift arithmetic right			x - - - x x u x x	98
	Register with single-shift	2	2		
	Register with variable-shift	8+4/bit	2		
	Memory with single-shift	15+EA	2–4		
	Memory with variable-shift	20+EA +4/bit	2–4		
SBB	Subtract with borrow			x - - - x x x x x	65
	Register from register	3	2		
	Memory from register	9+EA	2–4		
	Register from memory	16+EA	2–4		
	Immediate from accumulator	4	2–3		
	Immediate from register	4	3–4		
	Immediate from memory	17+EA	3–6		
SCAS/	Scan string/		1	x - - - x x x x x	208
SCASB/	Scan byte string/				
SCASW	Scan word string				
	Not repeated	15			
	Repeated	9+15/rep			
SHR	Shift logical right			x - - - x x u x x	98
	Register with single-shift	2	2		
	Register with variable-shift	8+4/bit	2		
	Memory with single-shift	15+EA	2–4		
	Memory with variable-shift	20+EA +4/bit	2–4		
STC	Set carry flag	2	1	- - - - - - - - 1	92
STD	Set direction flag	2	1	- 1 - - - - - - -	92

Mnemonic	Description	Clock cycles	Number of bytes	Flags O D I T S Z A P C	Page ref.
STI	Set interrupt flag	2	1	- - 1 - - - - - -	92
STOS/	Store string/		1	- - - - - - - - - -	208
STOSB/	Store byte string/				
STOSW	Store word string				
	Not repeated	11			
	Repeated	9+10/rep			
SUB	Subtraction			x - - - x x x x x	65
	Register from register	3	2		
	Memory from register	9+EA	2–4		
	Register from memory	16+EA	2–4		
	Immediate from accumulator	4	2–3		
	Immediate from register	4	3–4		
	Immediate from memory	17+EA	3–6		
TEST	Test			0 - - - x x u x 0	93
	Register with register	3	2		
	Memory with register	9+EA	2–4		
	Immediate with accumulator	4	2–3		
	Immediate with register	5	3–4		
	Immediate with memory	11+EA	3–6		
WAIT	Wait while $\overline{\text{TEST}}$ pin not asserted	3+5n	1	- - - - - - - - - -	457
XCHG	Exchange			- - - - - - - - - -	60
	Register with accumulator	3	1		
	Register with memory	17+EA	2–4		
	Register with register	4	2		
XLAT/ XLATB	Translate	11	1	- - - - - - - - - -	221
XOR	Logical exclusive OR			0 - - - x x u x 0	93
	Register with register	3	2		
	Memory with register	9+EA	2–4		
	Register with memory	16+EA	2–4		
	Immediate with accumulator	4	2–3		
	Immediate with register	4	3–4		
	Immediate with memory	17+EA	3–6		

EA	No. of Clock Cycles
Direct	6
Register indirect	5
Register relative	9
Based indexed	
(BP)+(DI) or (BX)+(SI)	7
(BP)+(SI) or (BX)+(DI)	8
Based indexed relative	
(BP)+(DI)+DISP or (BX)+(SI)+DISP	11
(BP)+(SI)+DISP or (BX)+(DI)+DISP	12

Flag Setting Symbols:

O D I T S Z A P C:
- – Not affected
- x Set or cleared according to the result
- u Undefined
- 0 Cleared to 0
- 1 Set to 1
- r Restored from previously saved value

Conditional Jump Instructions

Instruction	Description	Condition	Aliases	Opposite
JC	Jump if carry	Carry = 1	JB, JNAE	JNC
JNC	Jump if no carry	Carry = 0	JNB, JAE	JC
JZ	Jump if zero	Zero = 1	JE	JNZ
JNZ	Jump if not zero	Zero = 0	JNE	JZ
JS	Jump if sign	Sign = 1	-	JNS
JNS	Jump if no sign	Sign = 0	-	JS
JO	Jump if overflow	Overflow = 1	-	JNO
JNO	Jump if no overflow	Overflow = 0	-	JO
JP	Jump if parity	Parity = 1	JPE	JNP
JPE	Jump if parity even	Parity = 1	JP	JPO
JNP	Jump if no parity	Parity = 0	JPO	JP
JPO	Jump if parity odd	Parity = 0	JNP	JPE

Unsigned Comparisons				
Instruction	Description	Condition	Aliases	Opposite
JA	Jump if above (>)	Carry = 0, Zero = 0	JNBE	JNA
JNBE	Jump if not below nor equal (not <=)	Carry = 0, Zero = 0	JA	JBE
JAE	Jump if above or equal (>=)	Carry = 0	JNC, JNB	JNAE
JNB	Jump if not below (not <)	Carry = 0	JNC, JAE	JB
JB	Jump if below (<)	Carry = 1	JC, JNAE	JNB
JNAE	Jump if not above nor equal (not >=)	Carry = 1	JC, JB	JAE
JBE	Jump if below or equal (<=)	Carry = 1 or Zero = 1	JNA	JNBE
JNA	Jump if not above (not >)	Carry = 1 or Zero = 1	JBE	JA
JE	Jump if equal (=)	Zero = 1	JZ	JNE
JNE	Jump if not equal ()	Zero = 0	JNZ	JE

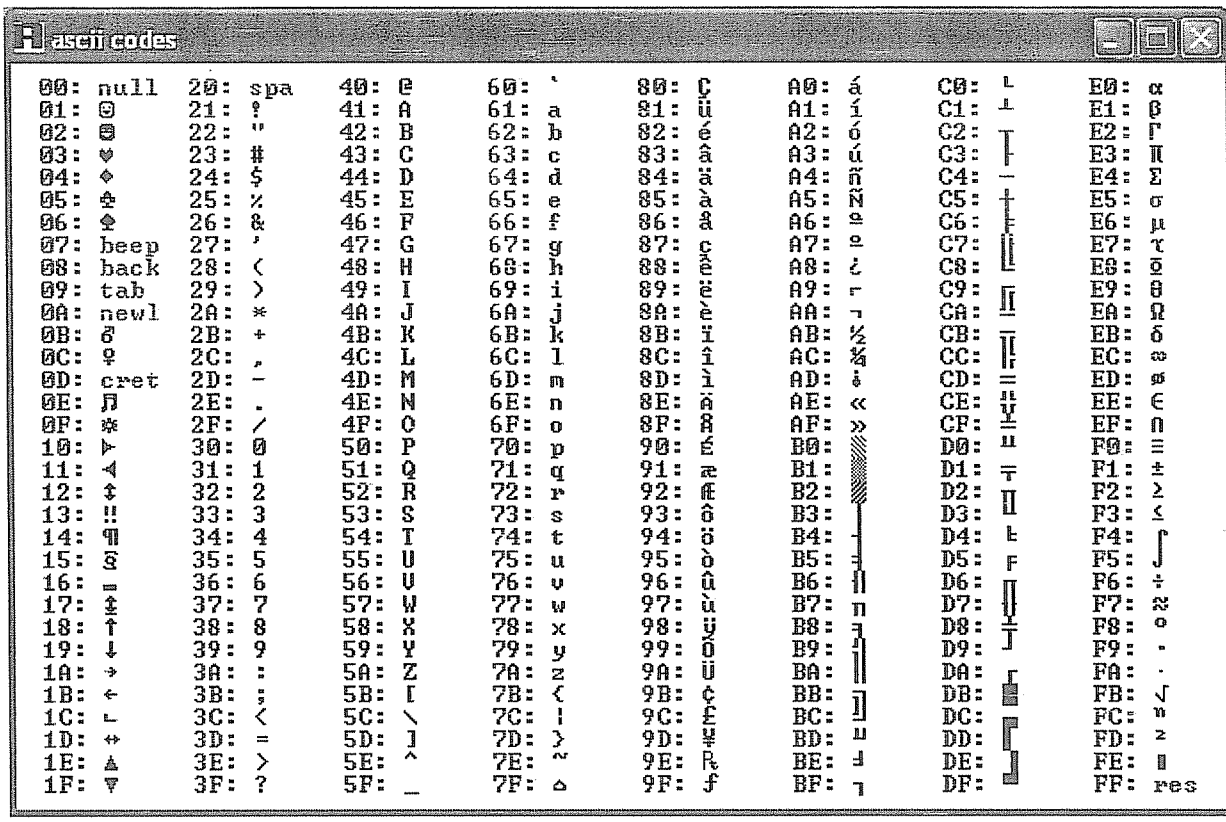
Signed Comparisons				
Instruction	Description	Condition	Aliases	Opposite
JG	Jump if greater (>)	Sign = Overflow or Zero = 0	JNLE	JNG
JNLE	Jump if not less than nor equal (not <=)	Sign = Overflow or Zero = 0	JG	JLE
JGE	Jump if greater than or equal (>=)	Sign = Overflow	JNL	JGE
JNL	Jump if not less than (not <)	Sign = Overflow	JGE	JL
JL	Jump if less than (<)	Sign Overflow	JNGE	JNL
JNGE	Jump if not greater nor equal (not >=)	Sign Overflow	JL	JGE
JLE	Jump if less than or equal (<=)	Sign Overflow or Zero = 1	JNG	JNLE
JNG	Jump if not greater than (not >)	Sign Overflow or Zero = 1	JLE	JG
JE	Jump if equal (=)	Zero = 1	JZ	JNE
JNE	Jump if not equal ()	Zero = 0	JNZ	JE

8086 Flags Register

```
|ODITSZAPC|     Overflow Flag, Direction Flag, Interrupt Flag, Trap Flag,
                Sign Flag, Zero Flag, Auxilary carry Flag, Parity Flag,
                Carry Flag
```

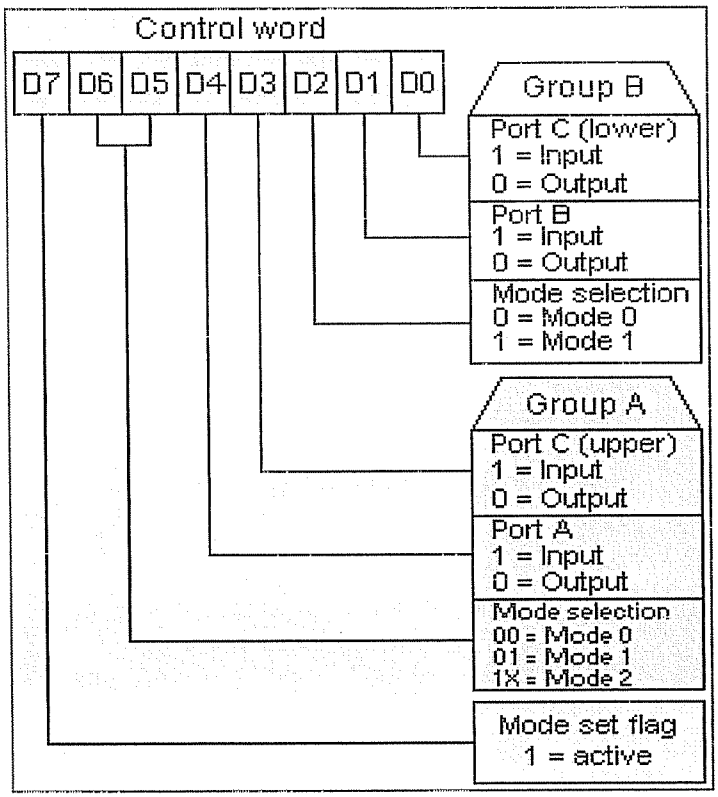
APPENDIX B: ASCII TABLE

ASCII Codes

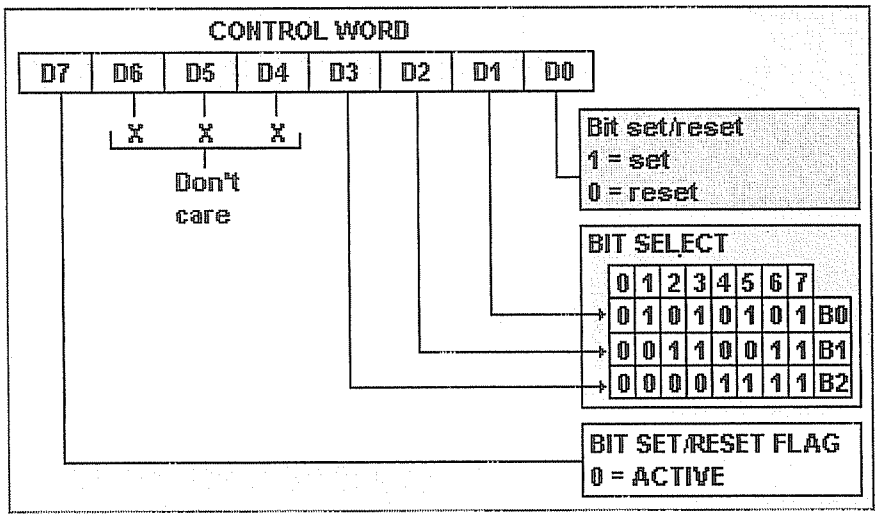


APPENDIX C: 8255 PPI

Control Word



8255 Control Word Format (I/O Mode)



BSR Control Word

APPENDIX D: 8253/8254 PIT

Control Word

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

BCD	Function
0	16-bit binary
1	4-decade BCD

SC1	SC0	Function
0	0	Select counter 0
0	1	Select counter 1
1	0	Select counter 2

RW1	RW0	Function
0	1	R/W LSB only
1	0	R/W MSB only
1	1	R/W LSB then MSB

M2	M1	M0	Mode
0	0	0	0
0	0	1	1
X	1	0	2
X	1	1	3
1	0	0	4
1	0	1	5