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

FINAL
Examination Paper

(COVER PAGE)

Session	:	**AUGUST 2014**
Programme	:	DIPLOMA IN ELECTRICAL AND ELECTRONIC ENGINEERING
Course	:	**EEE 2105: INTRODUCTION TO MICROPROCESSOR**
Date of Examination	:	December 9, 2014 (Tuesday)
Time	:	5.00pm – 7.00pm 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.

Students are also not allowed to write anything on the Appendix handout given.

Materials permitted :

Non-Programmable Scientific Calculator

Materials provided :

Appendix A (8086 Instruction Set Summary), Appendix B (ASCII Table), Appendix C (8255 PPI), Appendix D (8253/8254 PIT), Appendix E (8259 PIC)

Examiner(s)	:	**Mr. Steven Khoo**
Moderator	:	**Dr. Mandeep Singh**

This paper consists of 11 printed pages, including the cover page.

INTI INTERNATIONAL COLLEGE PENANG

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

EEE2105: INTRODUCTION TO MICROPROCESSORS
FINAL EXAMINATION: AUG2014 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) The following is output from the -r **DEBUG** commands after a certain 8086 program has run with a breakpoint set. All values are in hexadecimal.

```
-r
AX=C846  BX=D184  CX=0020  DX=017C  SP=0215  BP=0403  SI=7000  DI=8000
DS=2617  ES=12E4  SS=5487  CS=3C1A  IP=0108   NV UP EI PL NZ NA PO NC
5B4A:010D 30D7          ADC     BH, DL
```

(i) Provide the logical address of the next instruction to be executed.

(1 mark)

(ii) Provide the physical address of the next instruction to be executed.

(3 marks)

(iii) Identify the next instruction. Find the value of register `BX` and `IP` after the next instruction has been executed.

(6 marks)

(iv) Provide the status flags (Carry, Auxiliary Carry, Zero, Sign and Parity) condition after next instruction being executed according to output from -t **DEBUG** command.

(5 marks)

(b) Figure 1(b-1) shows a TM124MBK36E Dynamic RAM Module pin configuration. Figure 1(b-2) shows a TMS27PC240 FN UV Erasable PROM pin configuration.

Determine the following:

(i) total memory capacity from RAMs in Kbits,

(4 marks)

(ii) individual memory organization,

(2 marks)

(iii) number of address pins and number of data pins of each memory module.

(4 marks)

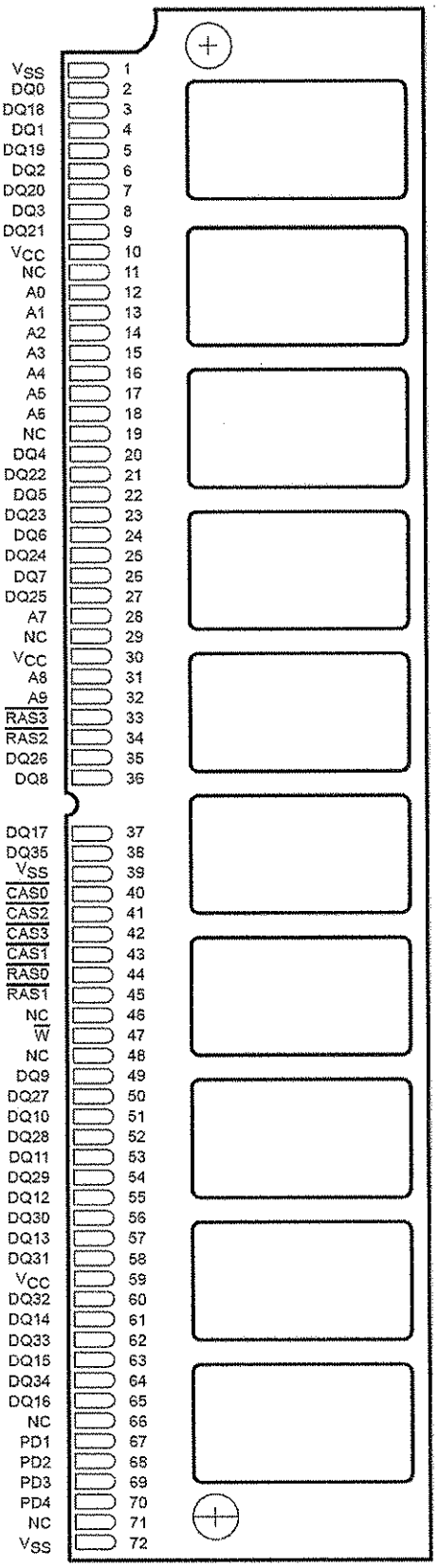


Figure 1(b-1) TM124MBK36E Dynamic RAM Module pin configuration.

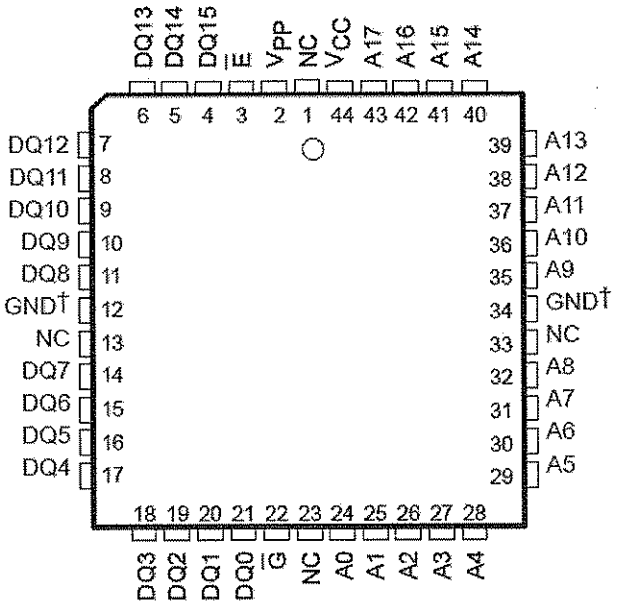


Figure 1(b-2) TMS27PC240 FN UV Erasable PROM pin configuration.

Question 2

Study and analyze the 8086 Assembly Language Codes below and answer the following questions. All values in the coding are in Hexadecimal.

Line 1	TITLE SAMPLE PROGRAM
Line 2	.MODEL SMALL
Line 3	.STACK 64
Line 4	.CODE
Line 5	.DATA
Line 6	MYCODE PROC
Line 7	MOV AX, 04
Line 8	MOV BX, 07DE
Line 9	CMP AX, BX
Line 10	JG testing1
Line 11	JLE testing
Line 12	testing1:
Line 13	ADD AL, 30
Line 14	JMP exit
Line 15	testing2:
Line 16	ADD BL, 20
Line 17	JMP finish
Line 18	finish:
Line 19	MOV AH, 4CL
Line 20	INT 21
Line 21	MYCODE ENDP
Line 22	END MYCODE

Table 2 Coding

(a) Identify THREE (3) errors in the above instructions. Briefly explain why it is incorrect and write the correct codes according to the Assembly Language.

(9 marks)

(b) What is the meaning of the instruction in Line 10 and 11? Can these instructions be reduced?

(6 marks)

(c) What is the meaning of the combined instructions at lines 18, 19 and 20?

(6 marks)

(d) Based on the assumption that all lines of codes are corrected, what is the final outcome of the above program?

(4 marks)

Question 3

(a) Calculate the address range for selecting the chip shown below in Figure 3(a).

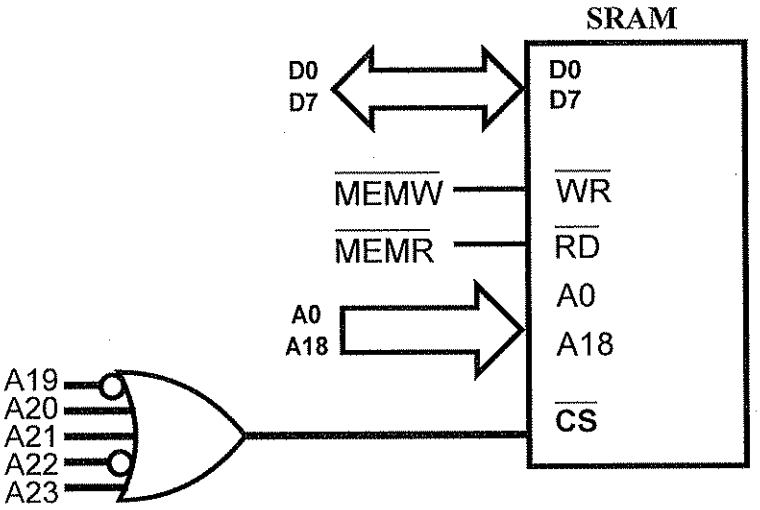


Figure 3(a) Address Decoding

(4 marks)

(b) Write an 8086 program that find out the largest number from a set of TEN 8-bit numbers stored in data segment of memory starting from address 1005H as shown below in Table 3(b). Store the result to address 100FH. Assume all ten data are unsigned numbers. The program length should be as minimum as possible.

(8 marks)

DS:	Offset	Data
0700H:	100FH	Result
0700H:	100EH	02H
0700H:	100DH	15H
0700H:	100CH	54H
0700H:	100BH	89H
0700H:	100AH	FEH
0700H:	1009H	01H
0700H:	1008H	78H
0700H:	1007H	7FH
0700H:	1006H	80H
0700H:	1005H	67H

Table 3(b)

[Refer to Appendix A for 8086 instructions]

(c) The 8255 PPI is configured as shown in Figure 3(c).

(i) Find the port addresses and control register in Figure 3(c). Thus, program the PPI to set PC3 to high. Also include comments for any instruction used.

(5 marks)

(ii) Write a program to test the incoming data from at port C continuously via 8255 PPI. If the data is an odd number, send 55H to port B but if data is an even number, send AAH to port A. Also, include comments for any instruction used.

(8 marks)

Examples of incoming data and the respective action:

- 27_H (Odd Number) ⇨ send 55_H out to port B
- AA_H (Even Number) ⇨ send AA_H out to port A
- $C1_H$ (Odd Number) ⇨ send 55_H out to port B
- 86_H (Even Number) ⇨ send AA_H out to port A

[Refer to Appendix C for 8255 PPI Control Word]

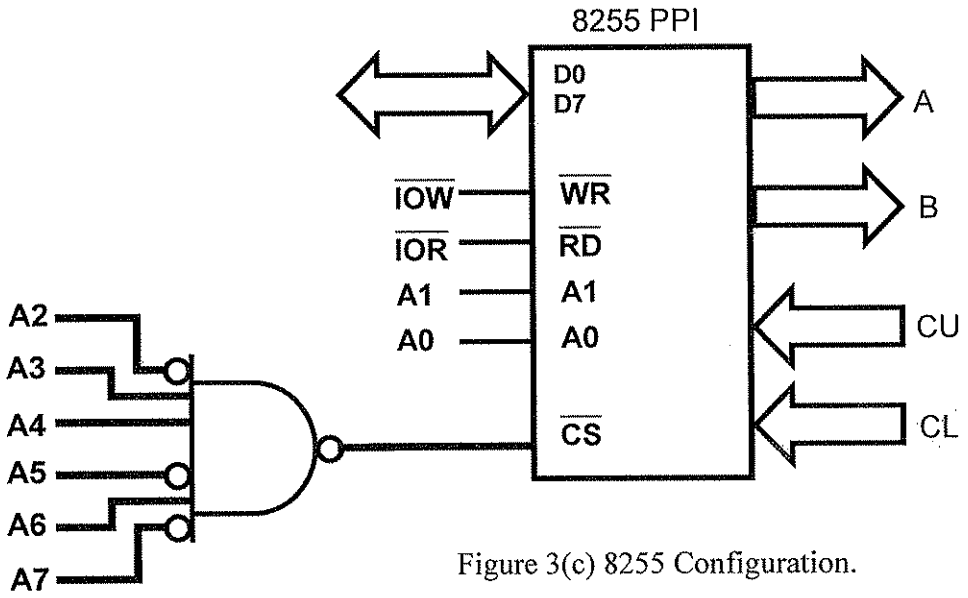


Figure 3(c) 8255 Configuration.

Question 4

(a) The 8259 PIC is configured as shown in Figure 4(a).

(i) Find the Initialization Control Words (ICWs) of the 8259 if it is used with an 8088/86 microprocessor, single, level triggering IRs, and IR1 is assigned "INT 38H". The 8259 is in master buffered mode with auto EOI and special fully nested.

(6 marks)

(ii) Write a program to initialize the 8259 using the port addresses in Figure 4(a). Also include comments for any instruction used.

(6 marks)

[Refer to Appendix E for 8259 PIC Control Words]

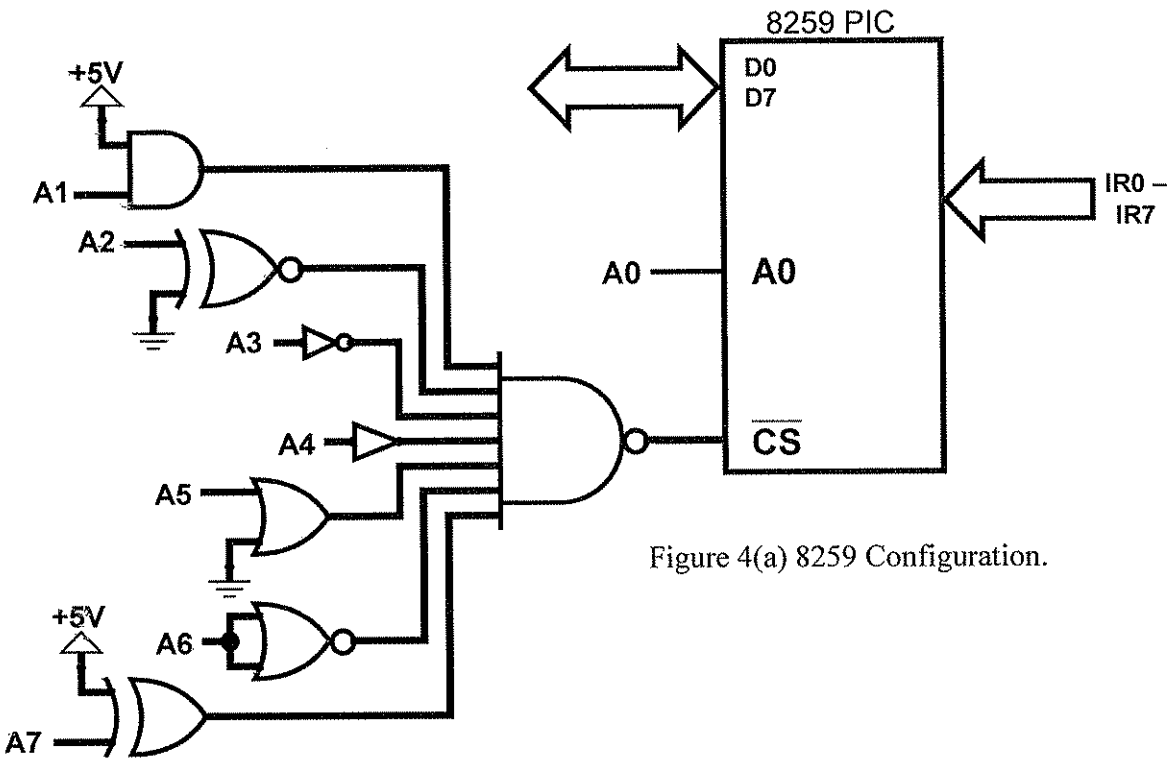


Figure 4(a) 8259 Configuration.

(b) Calculate the time delay taken for Program Q4(b) running on 8086 microprocessor at 10MHz. During the state of executing Program Q4(b), the RAM content in Table 4(b) is as follows: (All values are in Hexadecimal). Assume that DS register contains 0700H. Show all workings clearly for each instruction.

(13 marks)

[Refer to Appendix A for the cycle time]

	Memory															
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
07700H	08	04	45	78	10	21	22	33	34	25	26	27	28	49	40	14
07800H	05	03	02	01	12	24	56	88	91	13	34	56	98	50	19	62
07900H	07	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0	00
07A00H	1F	1E	1D	1C	1B	1A	55	66	47	46	45	44	43	42	41	0F
07B00H	18	19	23	48	25	26	27	58	29	31	32	35	36	37	38	39

Table 4(b)

Program		
	MOV	AL, 01H
LOOP3:	MOV	BL, [0A03H]
LOOP2:	MOV	CX, [0802H]
LOOP1:	LOOP	LOOP1
	DEC	BL
	JNZ	LOOP2
	NOP	
	DEC	AL
	JNZ	LOOP3
	HLT	

Program Q4(b)

Question 5

(a) The 8237 DMA is configured as shown in Figure 5(a). Determine the port addresses assigned to the four channels. Write a program using channel 1 memory address and count registers to transfer 2K starting from offset 6789H using the port address obtained. (5 marks)

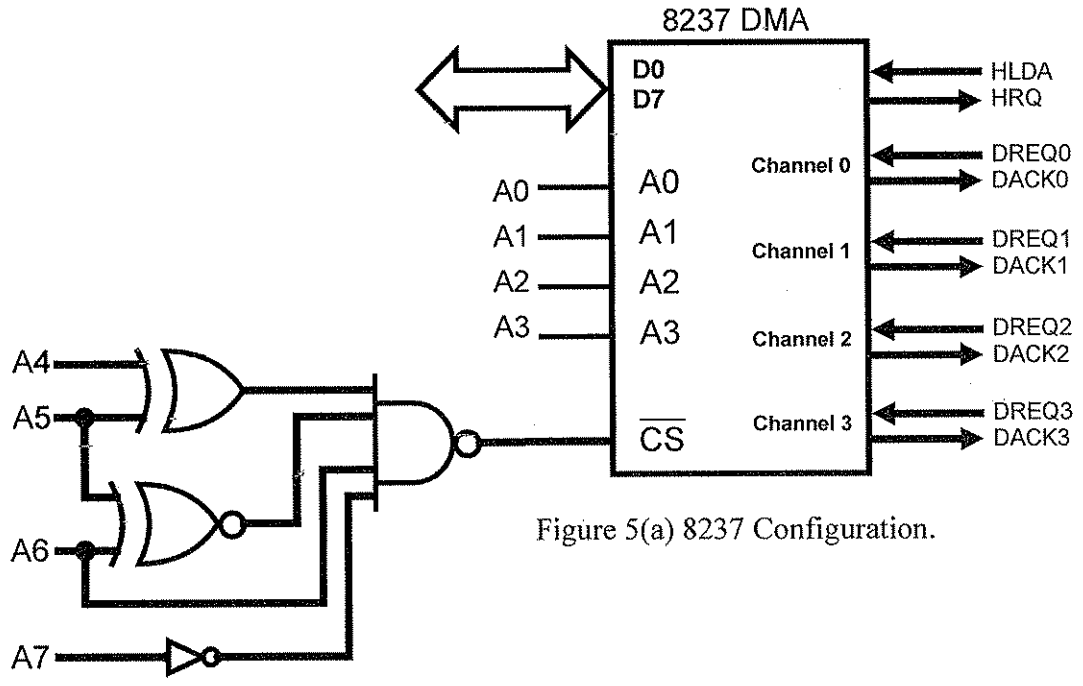
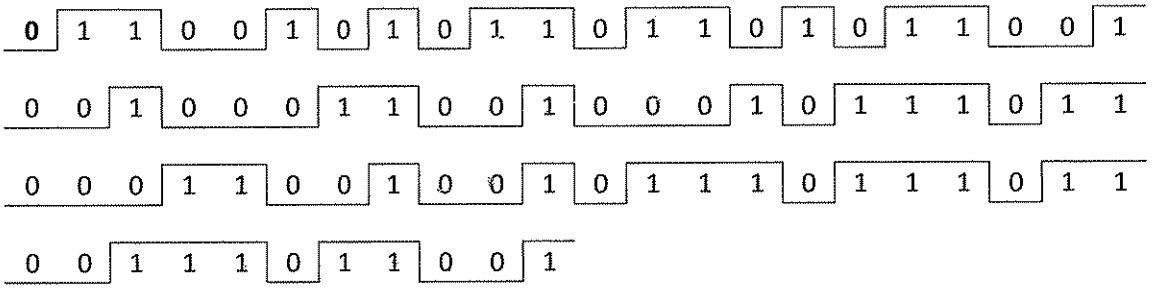


Figure 5(a) 8237 Configuration.

(b) A transmission system uses asynchronous serial data communication as shown in Figure 5(b) with **LSB** being transmitted first to transmit a passkey to the receiver.

Decode the following serial data received in continuous ASCII characters message:

Incoming data:



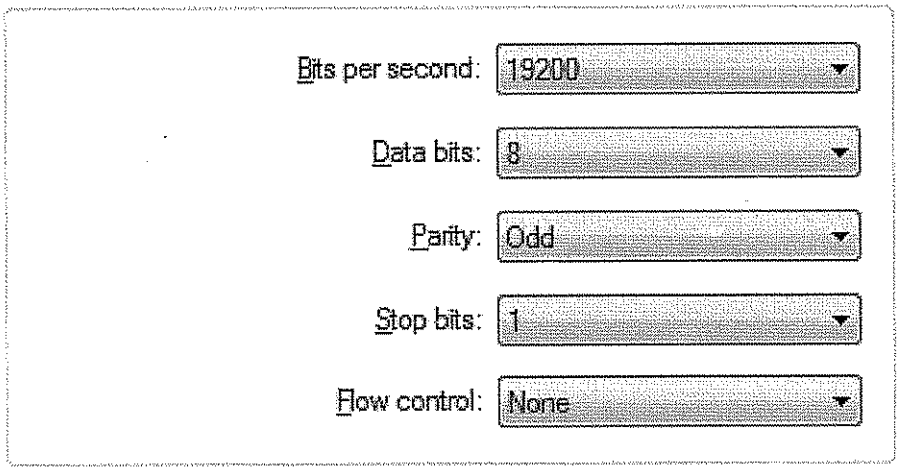


Figure 5(b) Serial communication settings.

(i) What is the passkey transmitted?

(8 marks)

(ii) Calculate the total time wasted due to overhead when transmitting the above message.

(3 marks)

[Refer to Appendix B for ASCII codes]

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

i) 2014.2014_8 to decimal equivalent.

(3 marks)

ii) 2014.2014_{10} to hexadecimal equivalent.

(3 marks)

iii) $[0100\ 0110\ 0100_2 + 0011\ 0011\ 0101_2]$ to BCD equivalent.

(3 marks)

Question 6

(a) Comment on the error if any, in the following assembly language mnemonics. Otherwise explain the outcome of the operation.

i) IN 49H, AL

(2 marks)

ii) XCHG [13H], [25H]

(2 marks)

(b) Suppose that AX = 3024H, BX = 20CCH, CX = 8250H and SP = 0302H.

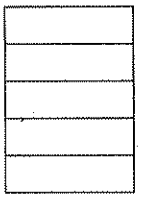
Determine the value of SP and the values of the data in the registers concerned as we progress through the following instructions. Show step-by-step after each instruction being executed using the aid of diagram as shown below to start with.

(10 marks)

```
PUSH CX
PUSH BX
PUSH
PUSH
POP
POP
POP
PUSH
POP
POP
```

Stack Memory

SS: 0300H
SS: 0301H
SS: 0302H ⇨
SS: 0303H
SS: 0304H



START

SP = 0302H

(c) The 8254 PIT chip is configured as shown in Figure 6(c).

(i) Find the port address assigned to all the counters and the control register.

(2 marks)

(ii) CLK1 of counter 1 is 1.19318MHz and Gate1 is connected to high permanently. OUT1 of counter 1 is connected to IR0 (the highest-priority interrupt) of the 8259 interrupt controller to provide time-of-day interrupt (TOD) interrupt. Counter 1 uses a square wave to trigger the 8259. Write assembly instructions to divide this counter by 2043H. What is the OUT1 frequency?

(4 marks)

(iii) CLK2 of counter 2 is 1.19318MHz and Gate2 is connected to high permanently. Counter 2 generates a periodic pulse every 30.176μs to refresh DRAM memory of the computer. Write assembly instructions to generate this periodic pulse through OUT2.

(5 marks)

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

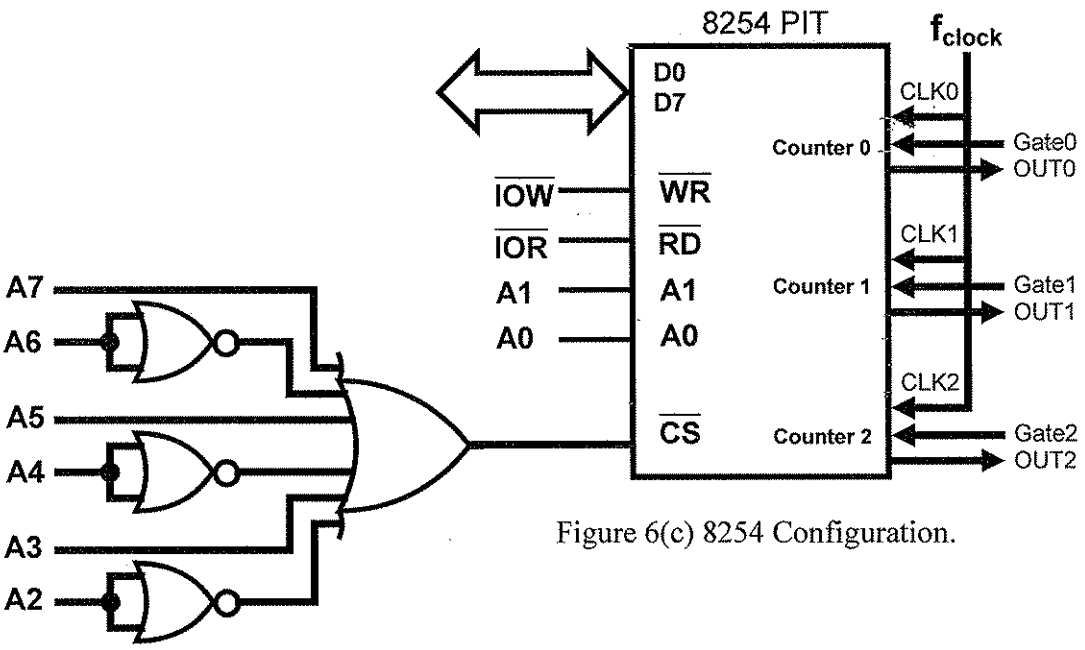


Figure 6(c) 8254 Configuration.

– THE END –

EEE2105(F)/Aug14/Steven Khoo/7/08/14

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) (BP)+(SI) or (BX)+(DI)	7 8
Based indexed relative (BP)+(DI)+DISP or (BX)+(SI)+DISP (BP)+(SI)+DISP or (BX)+(DI)+DISP	11 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

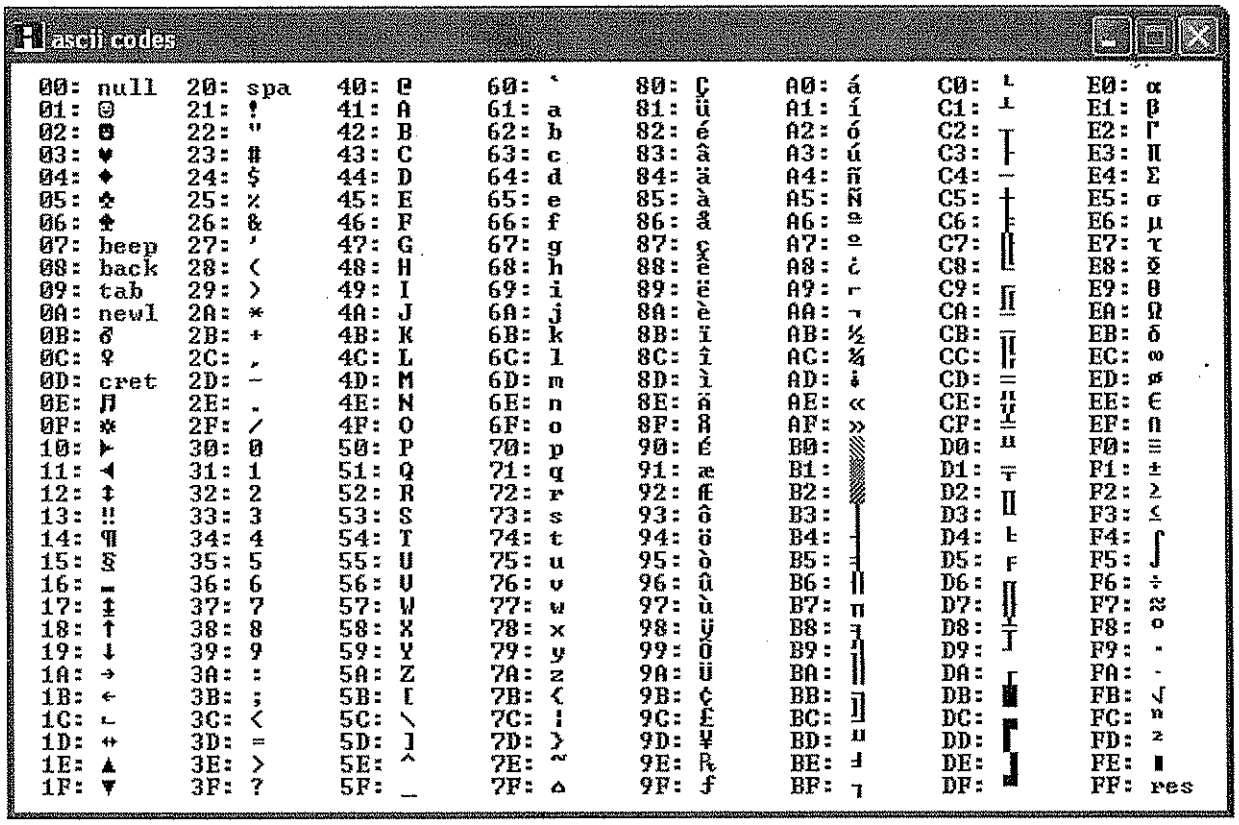
Signed Comparisons				
Instruction	Description	Condition	Aliases	Opposite
JG	Jump if greater (>)	Sign = Overflow or Zero = 0	JNLE	JNG
JNLE	Jump if not less than nor equal (not <=)	Sign = Overflow or Zero = 0	JG	JLE
JGE	Jump if greater than or equal (>=)	Sign = Overflow	JNL	JGE
JNL	Jump if not less than (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
```

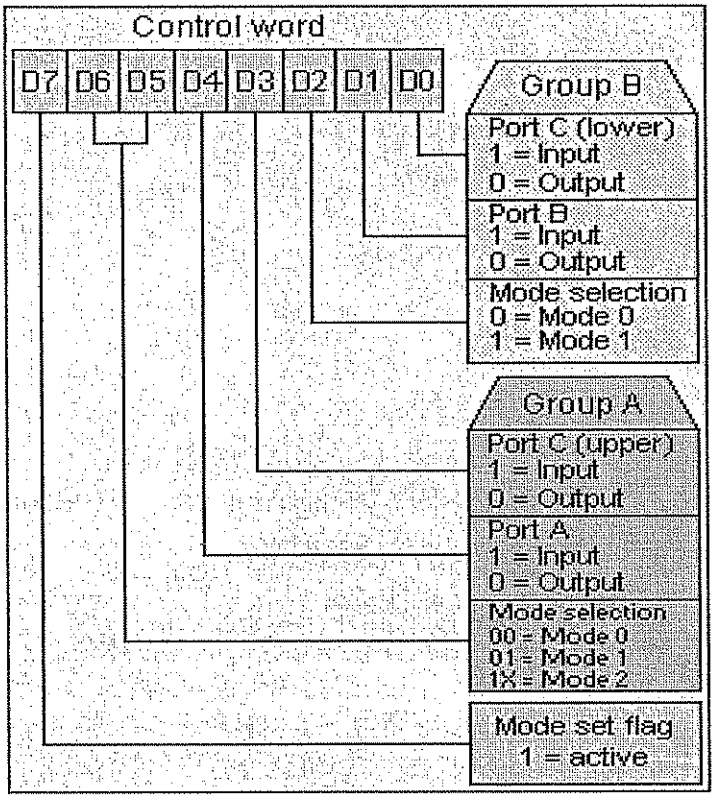
APPENDIX B: ASCII TABLE

ASCII Codes

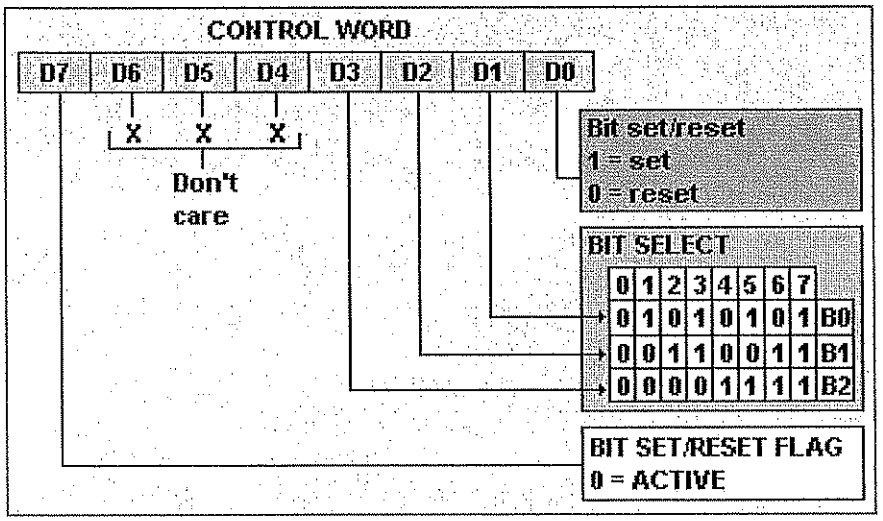


APPENDIX C: 8255 PPI

Control Word



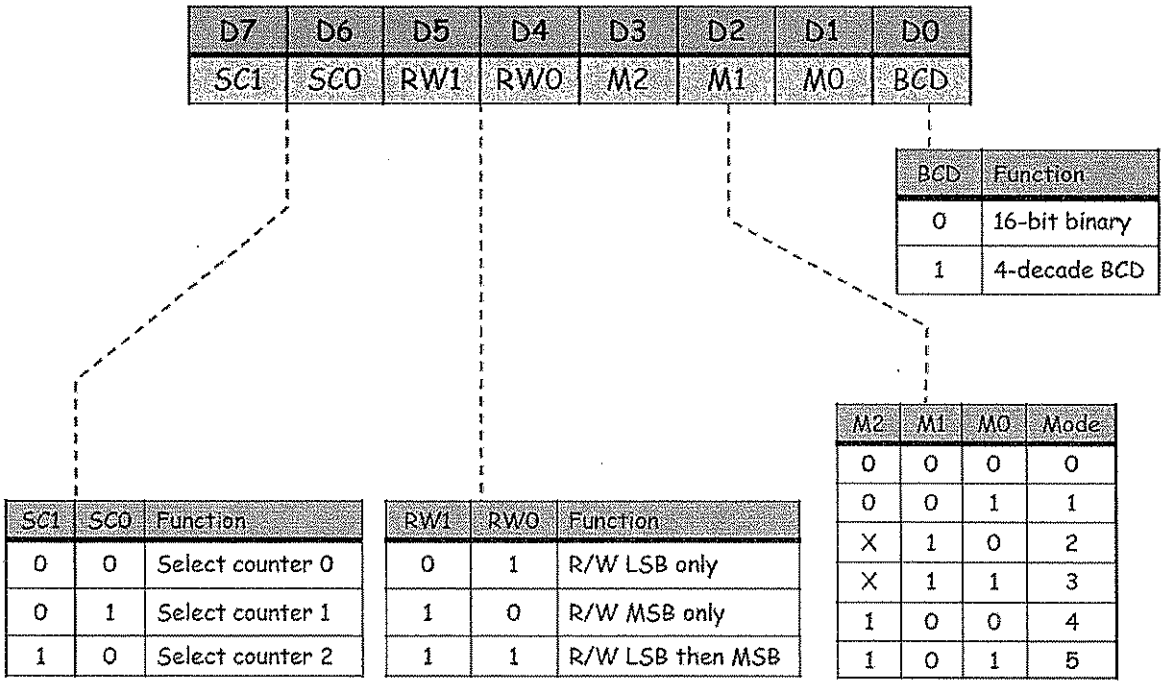
8255 Control Word Format (I/O Mode)



BSR Control Word

APPENDIX D: 8253/8254 PIT

Control Word



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

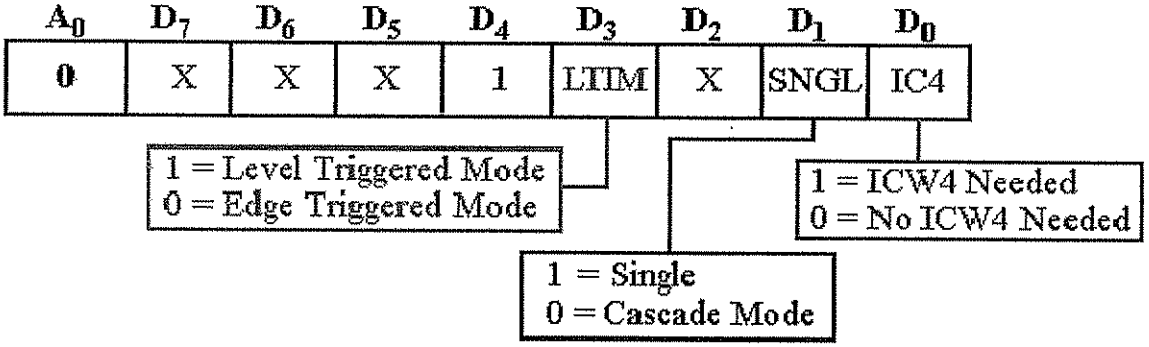
SC1	SC0	Function
0	0	Select counter 0
0	1	Select counter 1
1	0	Select counter 2

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

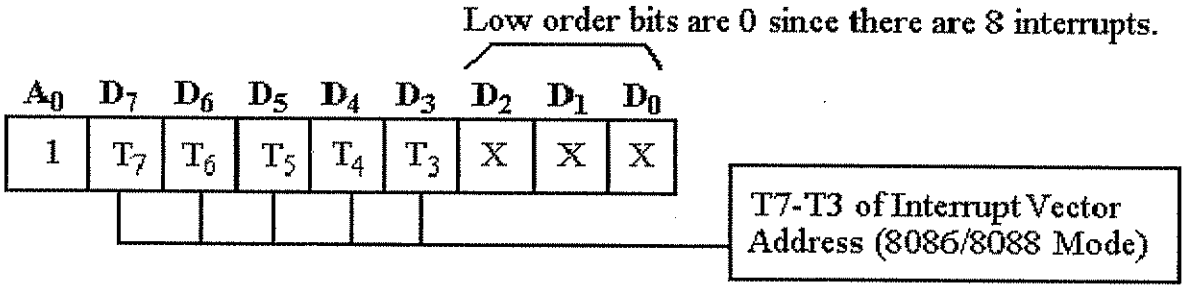
M2	M1	M0	Mode
0	0	0	0
0	0	1	1
X	1	0	2
X	1	1	3
1	0	0	4
1	0	1	5

APPENDIX E: 8259 PIC

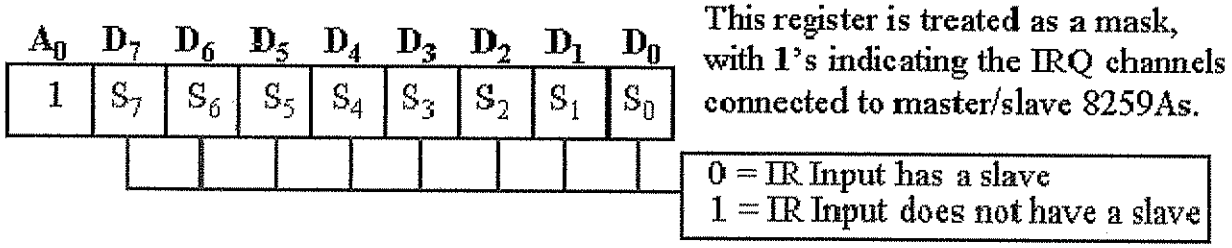
ICW1:



ICW2:



ICW3:



ICW4: