๒

Chapter2: Memory, I/O addressing

Asst.Prof.Dr.Supakit Nootyaskool

Objective

- Tell the relationship between binary and hexadecimal.
- Understand parity and error correction.
- Know what is inside the text file.
- Understand address bus, data bus, and control signal
- Tell devices selecting on the decoder circuit.

Topics

- Relationship between binary and hexadecimal
- Parity and error correction
- Text data
- Address bus, Data bus, and Control signal
- Decoder

RELATIONSHIP BETWEEN BINARY AND HEXADECIMAL

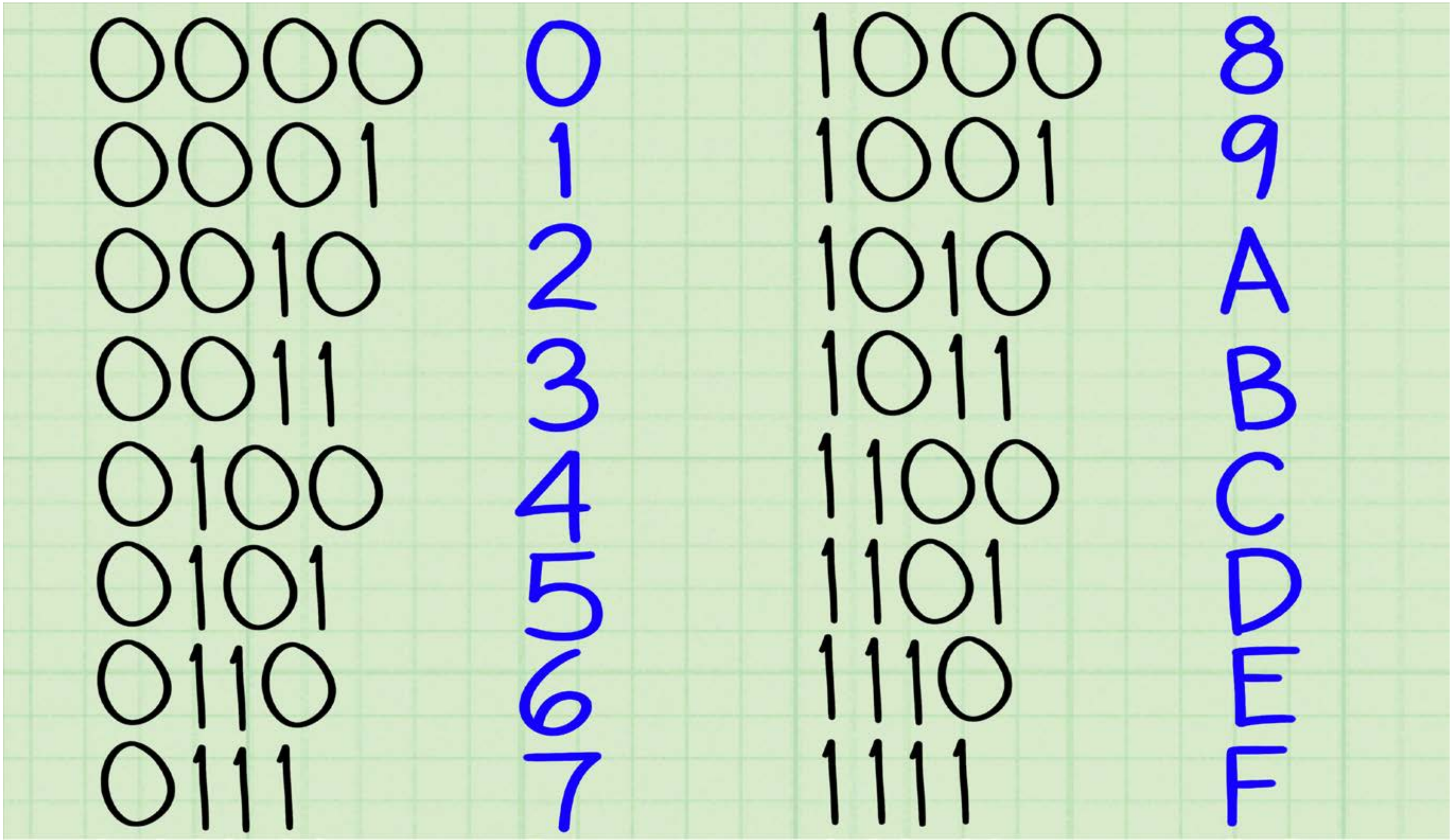
Able to convert binary to hexadecimal and convert back.

When you read a computer code

```
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#define PORT1 0x3F8  /* Defines Serial Port Base Address (COM1 */
void main(void){
    unsigned char c = 0;
    unsigned char chrctr = 0;
    /*int exit = 1; */
    outportb(PORT1 + 1, 0); /* Turn off interrupts */
    /* PORT1 Communication Settings */
    outportb(PORT1 + 3, 0x80); /* Set DLAB ON */
    outportb(PORT1 + 0, 0x0C); /* Set the baud rate to 9600  */
    outportb(PORT1 + 1, 0x00); /* Set Baud - Divisor latch HIGH */
    outportb(PORT1 + 3, 0x03); /* 8 bits, no parity, 1 stop */
    outportb(PORT1 + 2, 0xC7); /* FIFO Control Register */
    outportb(PORT1 + 4, 0x0B); /* Turn on DTR, RTS, and OUT2) */
    printf("Waiting on transmission from source.\nPress ESC to quit.\n");
    while(chrctr != 27){      /* Execute the loop if ESC has been hit */
    c = inportb(PORT1 + 5);
    if (c & 0x01){
        chrctr = inportb(PORT1);
        printf("%d",chrctr);
        }
    if (kbhit()){
        chrctr = getch();
        outportb(PORT1, chrctr);
        }
    }
    }"
```

What is 0x3F8?

Binary <-> Hexadecimal

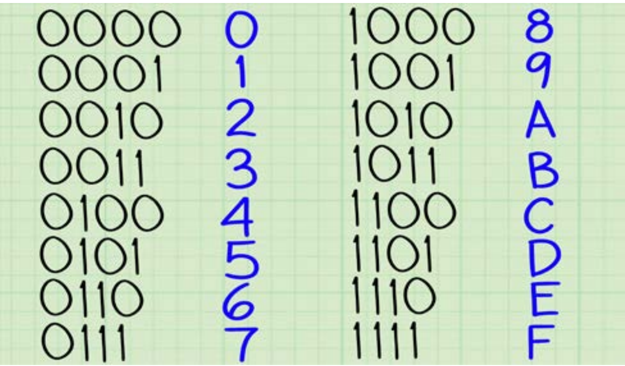


0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Activity2.1 Convert Binary and Hex

- 3F8H

  0011-_____-_____



0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

- 011011001110011110011$\underline{1101}$

  D

Format of the number system

- We can conclude that

$$\text{Value x Base}^{\text{position-1}}$$

- 101_2 = $1x2^1 + 0x2^1 + 1x2^0$ = $?_{10}$
- 453_8 = = $?_{10}$
- 453_{16} = = $?_{10}$

Relationship between binary, octal, and hexadecimal

- 4310_{10} = 1000011100001
  divided in 4 = 0001 0000 1110 0001
  divided in 3 = 001 000 011 100 001

0001 0000 1110 0001

Hex = 1 0 E 1 = $10E1_{16}$

001 000 011 100 001

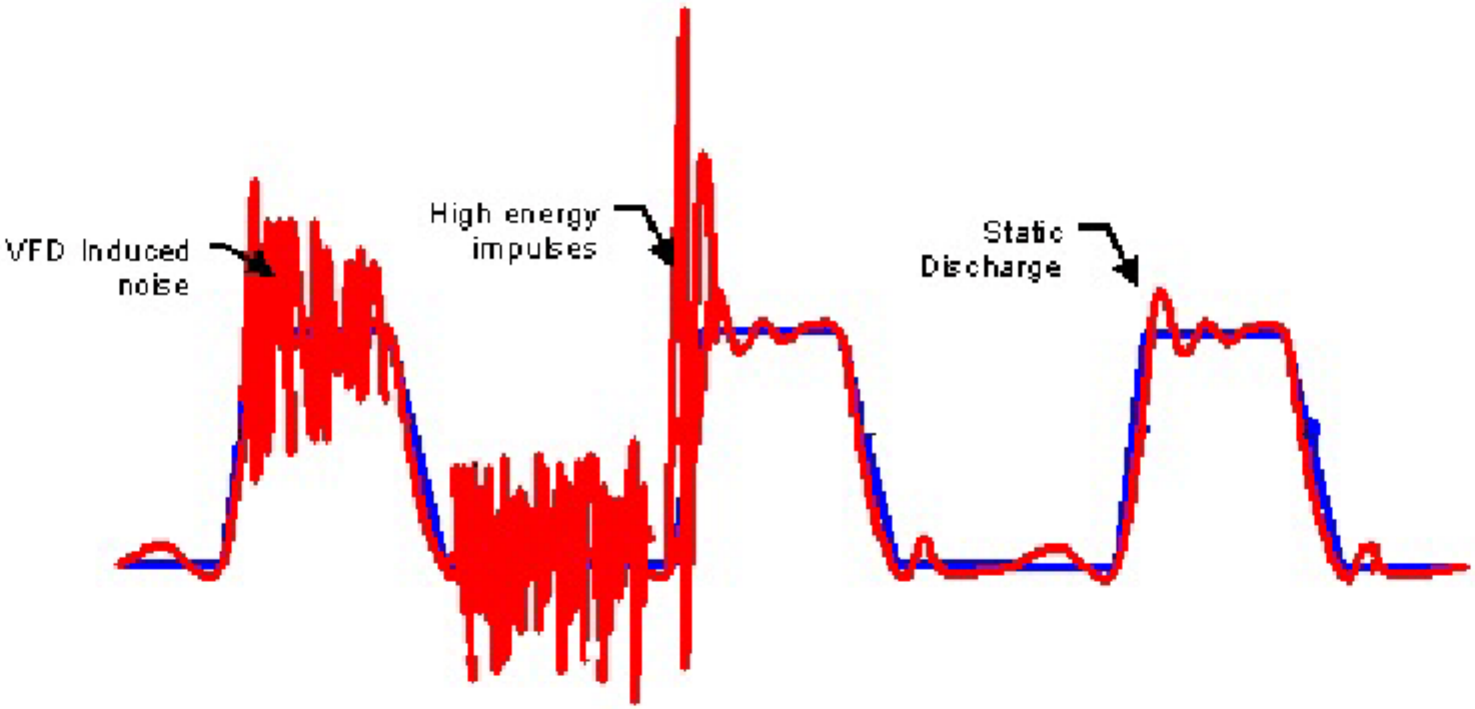
Oct = 1 0 3 4 1 = 10341_8

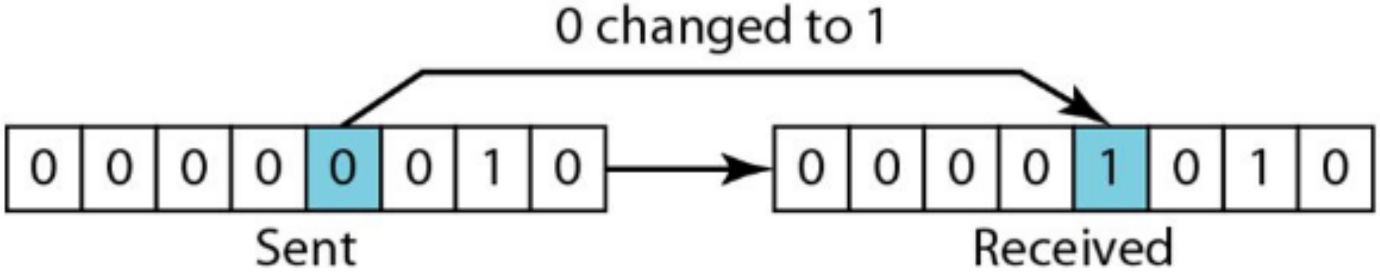
PARITY AND ERROR CORRECTION

Write odd and even parity

Tell error correction checking

Error in signal communication




0 changed to 1

0 0 0 0 0 0 1 0 → 0 0 0 0 1 0 1 0

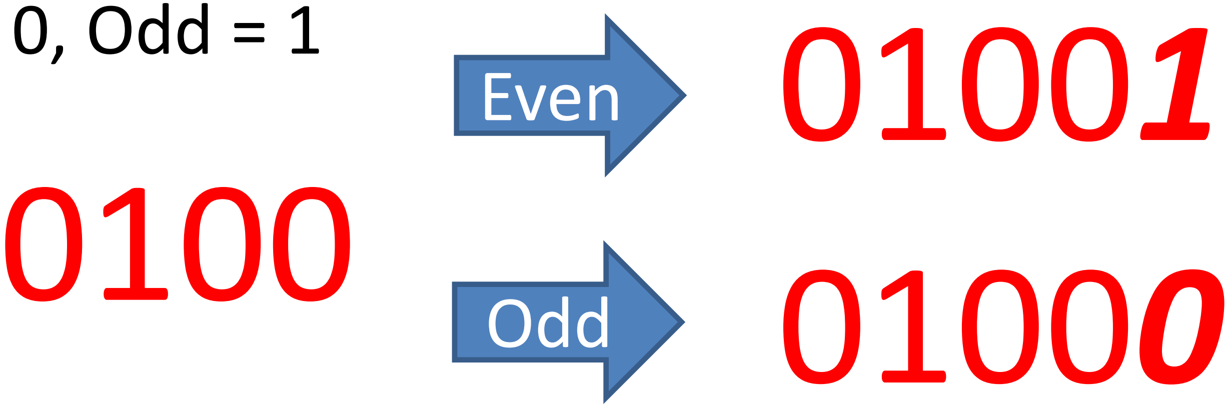
Sent

Received

Parity bit and error correction

- A parity bit is a bit appended to a binary sequence in order to sum of the number in even or odd.

- Even = 0, Odd = 1



0100

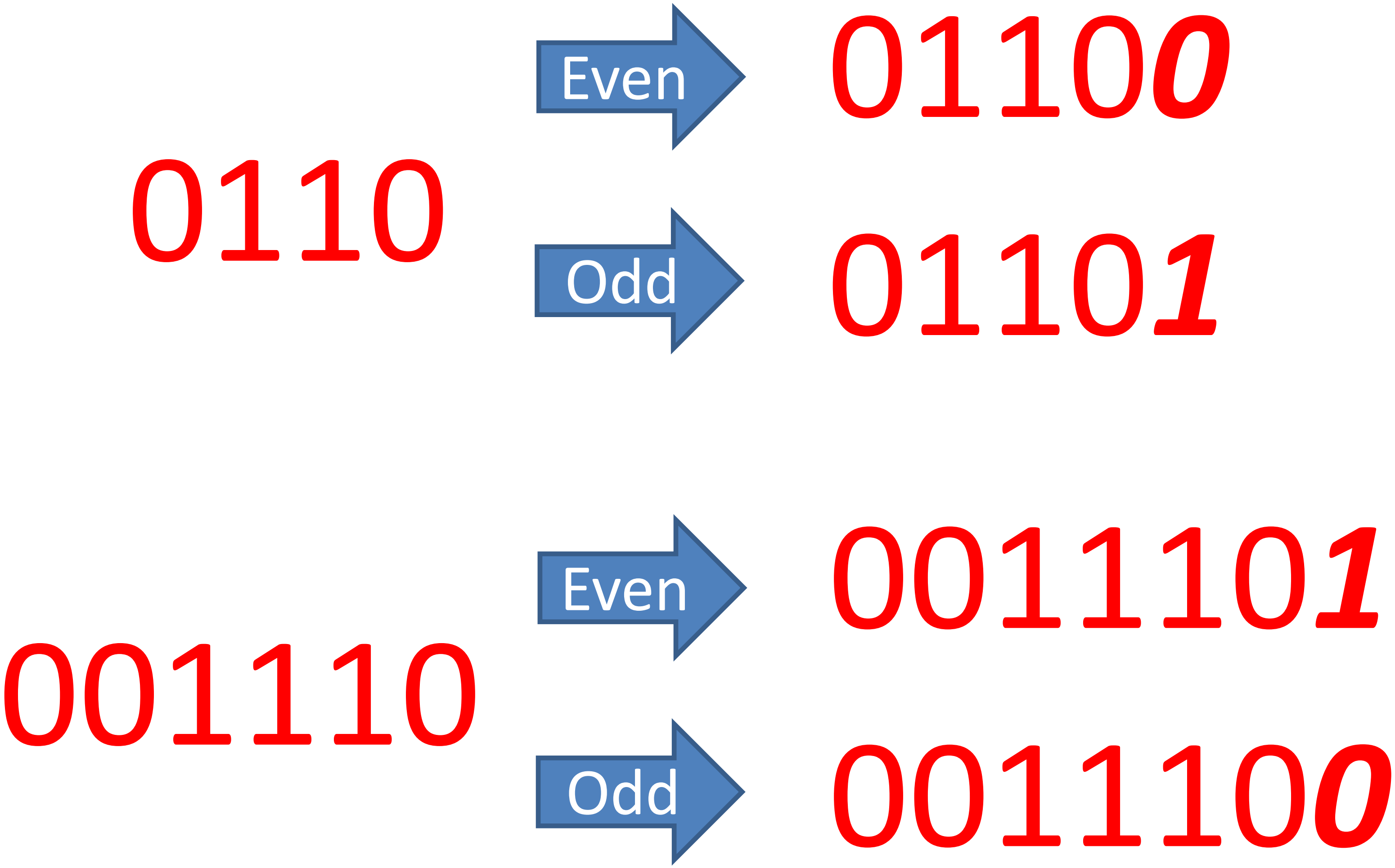
Even

0100***1***

Odd

0100***0***

Parity bit and error correction



0110

Even

0110***0***

Odd

0110***1***

001110

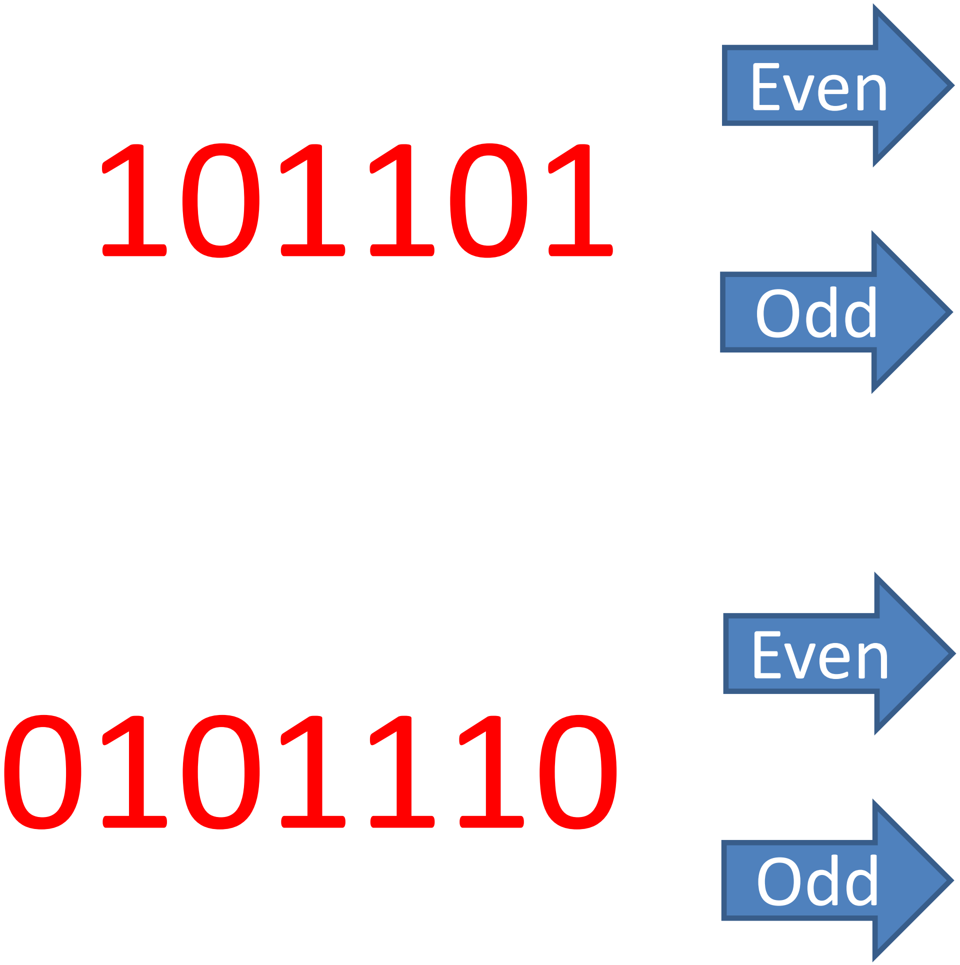
Even

001110***1***

Odd

001110***0***

Activity 2.2 write parity bit



101101

Even

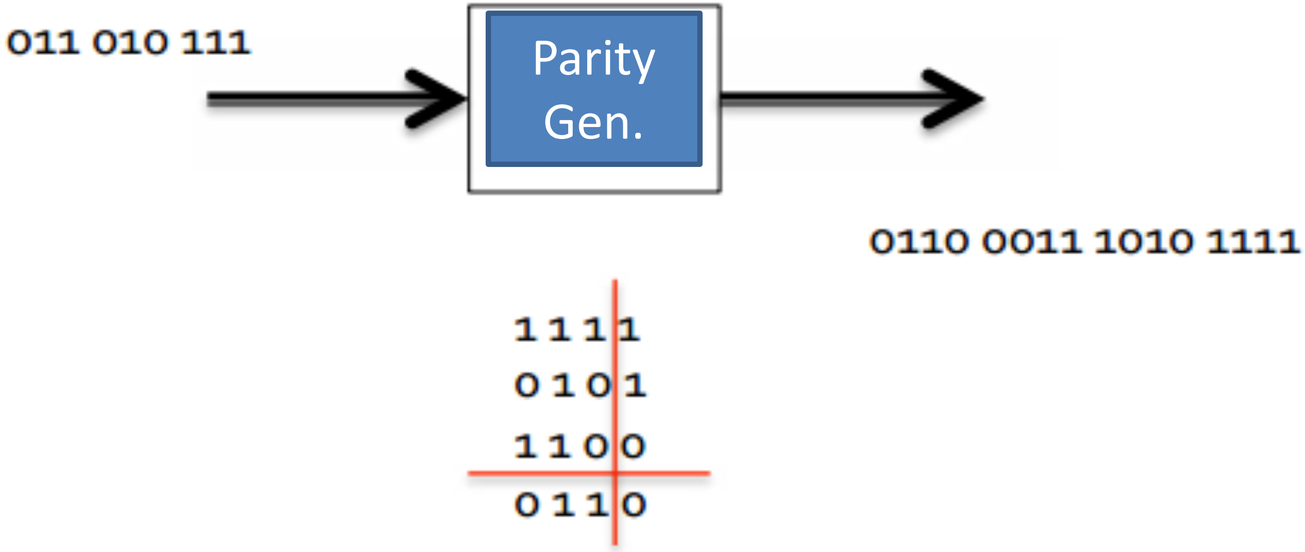
Odd

0101110

Even

Odd

Error correction



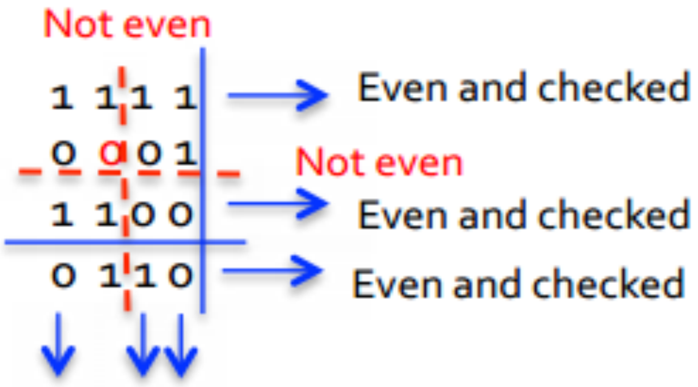
011 010 111

Parity Gen.

0110 0011 1010 1111

1 1 1 1
0 1 0 1
1 1 0 0
0 1 1 0

At receiver, if there is an error

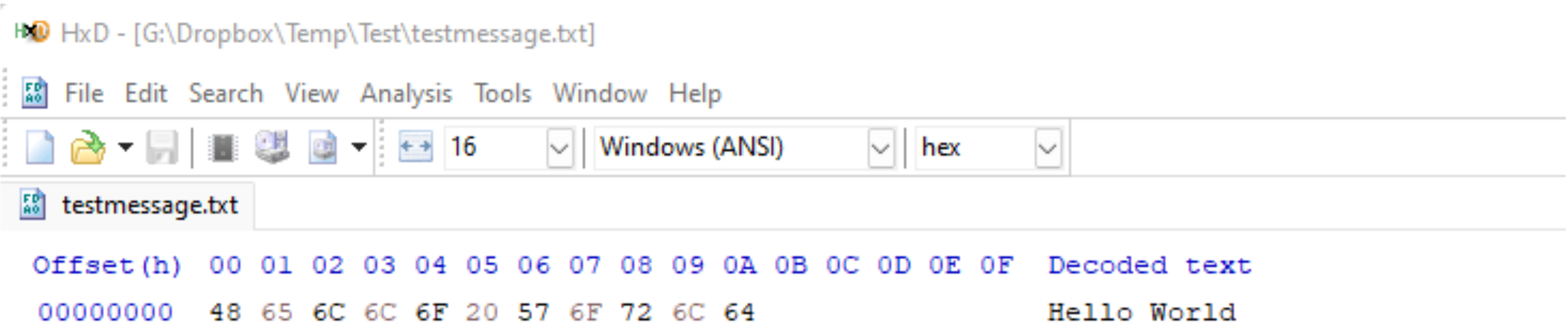


Not even

1 1 1 1 → Even and checked
0 0 0 1 Not even
1 1 0 0 → Even and checked
0 1 1 0 → Even and checked

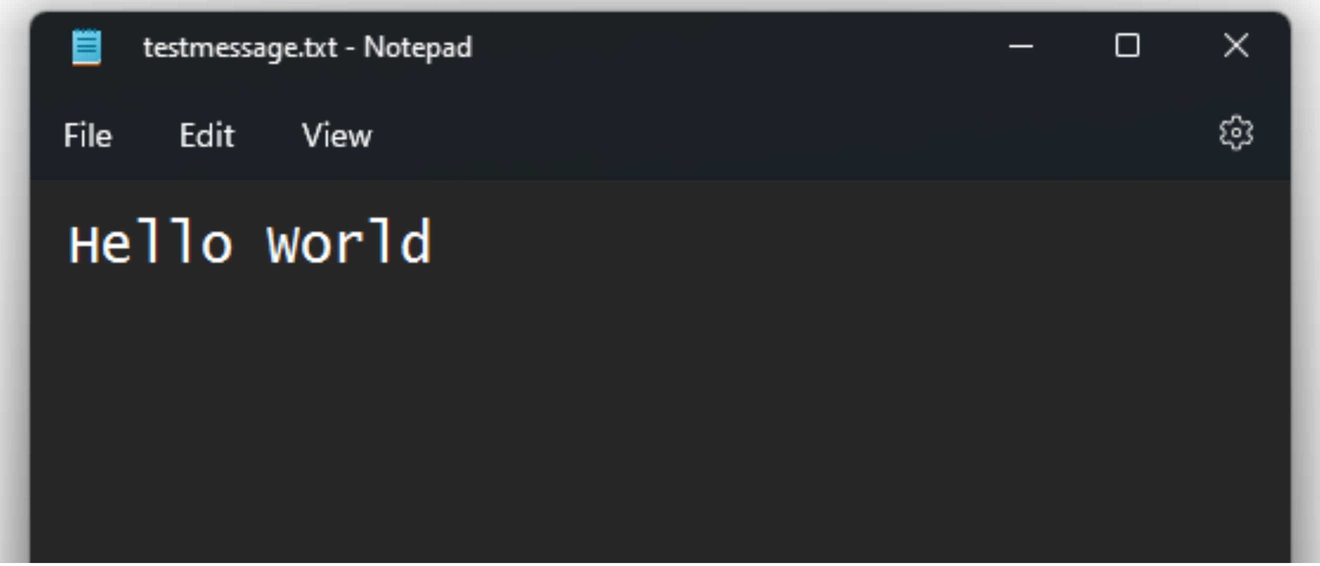
TEXT DATA

Text file


HxD - [G:\Dropbox\Temp\Test\testmessage.txt]
File Edit Search View Analysis Tools Window Help
testmessage.txt

16 | Windows (ANSI) | hex

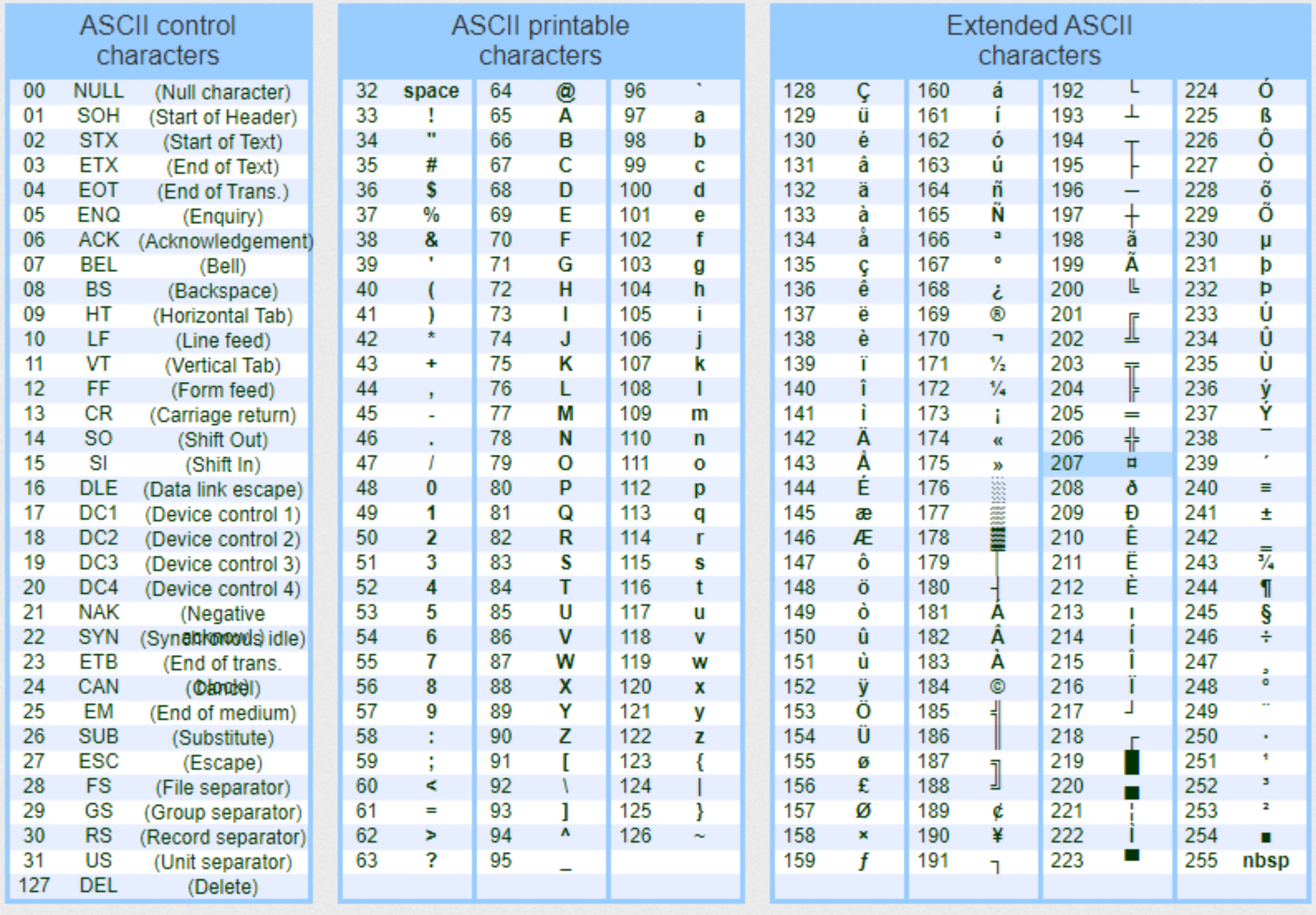
```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00000000  48 65 6C 6C 6F 20 57 6F 72 6C 64                 Hello World
```



```
Hello World
```

https://mh-nexus.de/en/

American Standard Code for Information Interchange (ASCII)

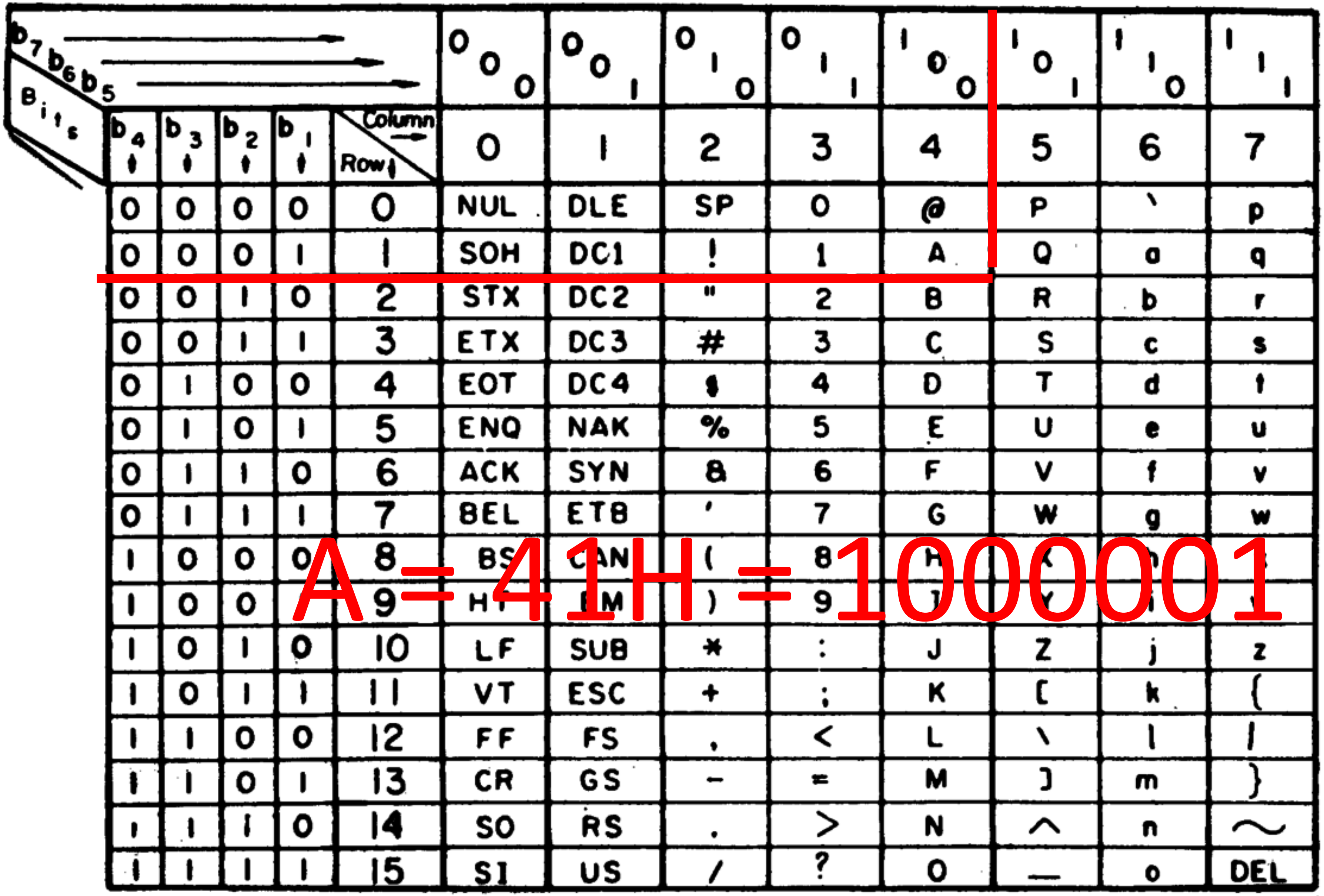


ASCII control characters			ASCII printable characters						Extended ASCII characters							
00	NULL	(Null character)	32	space	64	@	96	`	128	Ç	160	á	192	└	224	Ó
01	SOH	(Start of Header)	33	!	65	A	97	a	129	ü	161	í	193	┴	225	ß
02	STX	(Start of Text)	34	"	66	B	98	b	130	é	162	ó	194	┬	226	Ô
03	ETX	(End of Text)	35	#	67	C	99	c	131	â	163	ú	195	├	227	Ò
04	EOT	(End of Trans.)	36	$	68	D	100	d	132	ä	164	ñ	196	─	228	õ
05	ENQ	(Enquiry)	37	%	69	E	101	e	133	à	165	Ñ	197	┼	229	Õ
06	ACK	(Acknowledgement)	38	&	70	F	102	f	134	å	166	ª	198	ã	230	µ
07	BEL	(Bell)	39	'	71	G	103	g	135	ç	167	º	199	Ã	231	þ
08	BS	(Backspace)	40	(	72	H	104	h	136	ê	168	¿	200	╚	232	Þ
09	HT	(Horizontal Tab)	41	)	73	I	105	i	137	ë	169	®	201	╔	233	Ú
10	LF	(Line feed)	42	*	74	J	106	j	138	è	170	¬	202	╩	234	Û
11	VT	(Vertical Tab)	43	+	75	K	107	k	139	ï	171	½	203	╦	235	Ù
12	FF	(Form feed)	44	,	76	L	108	l	140	î	172	¼	204	╠	236	ý
13	CR	(Carriage return)	45	-	77	M	109	m	141	ì	173	¡	205	═	237	Ý
14	SO	(Shift Out)	46	.	78	N	110	n	142	Ä	174	«	206	╬	238	¯
15	SI	(Shift In)	47	/	79	O	111	o	143	Å	175	»	207	¤	239	´
16	DLE	(Data link escape)	48	0	80	P	112	p	144	É	176	░	208	ð	240	≡
17	DC1	(Device control 1)	49	1	81	Q	113	q	145	æ	177	▒	209	Ð	241	±
18	DC2	(Device control 2)	50	2	82	R	114	r	146	Æ	178	▓	210	Ê	242	‗
19	DC3	(Device control 3)	51	3	83	S	115	s	147	ô	179	│	211	Ë	243	¾
20	DC4	(Device control 4)	52	4	84	T	116	t	148	ö	180	┤	212	È	244	¶
21	NAK	(Negative acknowledge)	53	5	85	U	117	u	149	ò	181	Á	213	ı	245	§
22	SYN	(Synchronous idle)	54	6	86	V	118	v	150	û	182	Â	214	Í	246	÷
23	ETB	(End of trans. block)	55	7	87	W	119	w	151	ù	183	À	215	Î	247	¸
24	CAN	(Cancel)	56	8	88	X	120	x	152	ÿ	184	©	216	Ï	248	°
25	EM	(End of medium)	57	9	89	Y	121	y	153	Ö	185	╣	217	┘	249	¨
26	SUB	(Substitute)	58	:	90	Z	122	z	154	Ü	186	║	218	┌	250	·
27	ESC	(Escape)	59	;	91	[	123	{	155	ø	187	╗	219	█	251	¹
28	FS	(File separator)	60	<	92	\	124	|	156	£	188	╝	220	▄	252	³
29	GS	(Group separator)	61	=	93	]	125	}	157	Ø	189	¢	221	¦	253	²
30	RS	(Record separator)	62	>	94	^	126	~	158	×	190	¥	222	Ì	254	■
31	US	(Unit separator)	63	?	95	_			159	ƒ	191	┐	223	▀	255	nbsp
127	DEL	(Delete)														

USASCII code chart

b_7 b_6 b_5 → Bits					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4 ↓	b_3 ↓	b_2 ↓	b_1 ↓	Column → Row ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

USASCII code chart



b7 b6 b5 → Bits					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b4 ↓	b3 ↓	b2 ↓	b1 ↓	Column → Row ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

A= 41H = 1000001

ADDRESS BUS, DATA BUS AND CONTROL SIGNAL

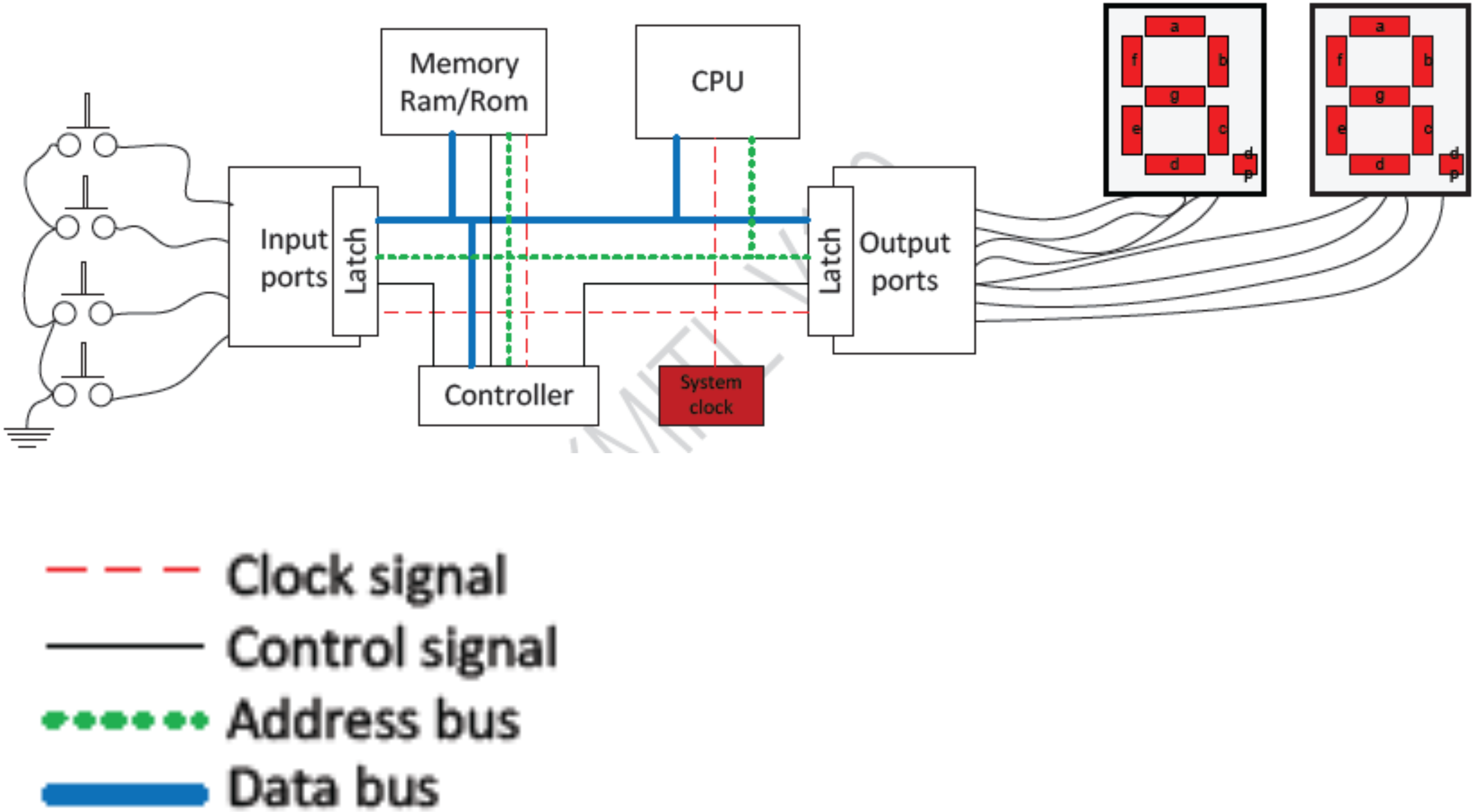
ADDRESS, DATA, and CONTROL

- **Address bus** is a sequence of binary data that points to the location of the data in the memory or I/O devices.
- **Control signal** is the signal sending to the devices such as the memory, the I/O to control reading, writing, enable, and disable.
- **Data bus** is the data that sending from a device to other device on the data wires. The data location relates to the address pointing the position.

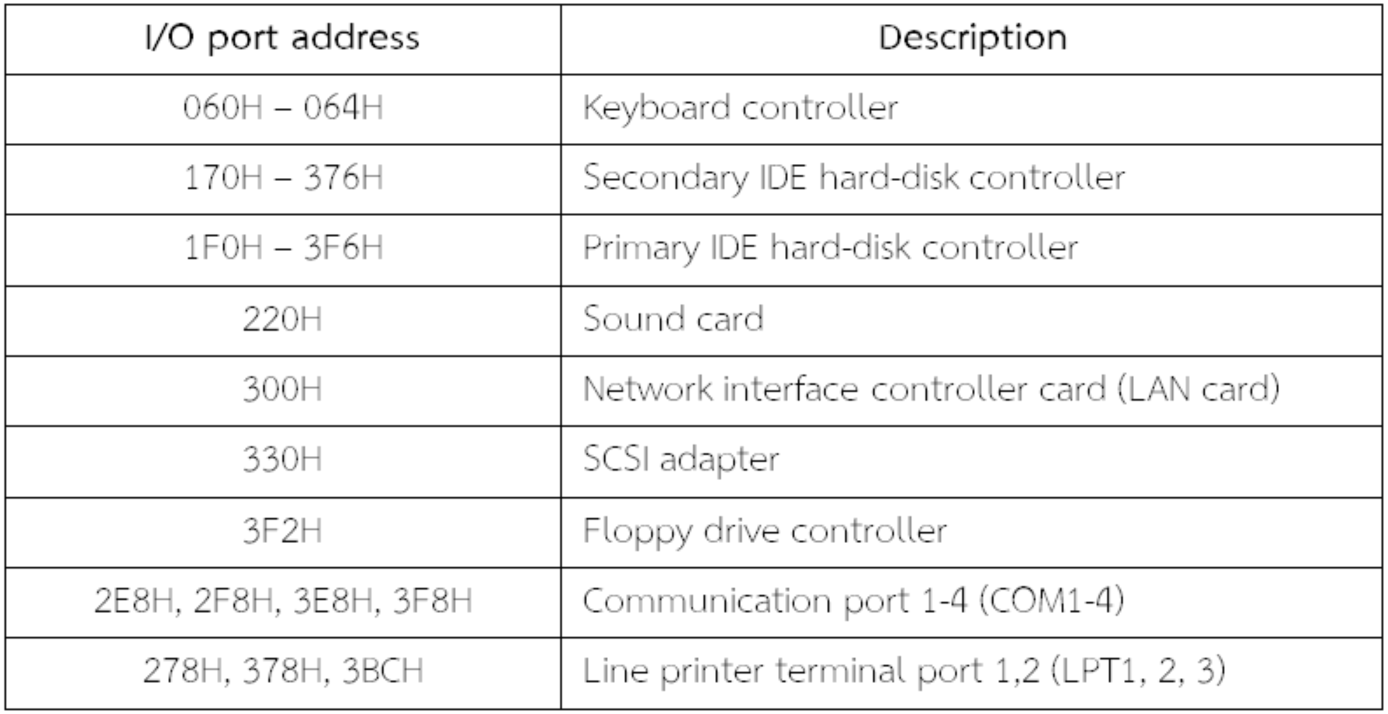
Activity 2.3 Describe process of parcel express.

- What is an issue concern in the parcel post?

Data transferring in the computer system



Device Address in computer



I/O port address	Description
060H – 064H	Keyboard controller
170H – 376H	Secondary IDE hard-disk controller
1F0H – 3F6H	Primary IDE hard-disk controller
220H	Sound card
300H	Network interface controller card (LAN card)
330H	SCSI adapter
3F2H	Floppy drive controller
2E8H, 2F8H, 3E8H, 3F8H	Communication port 1-4 (COM1-4)
278H, 378H, 3BCH	Line printer terminal port 1,2 (LPT1, 2, 3)

Example transferring data in the comport with C-language

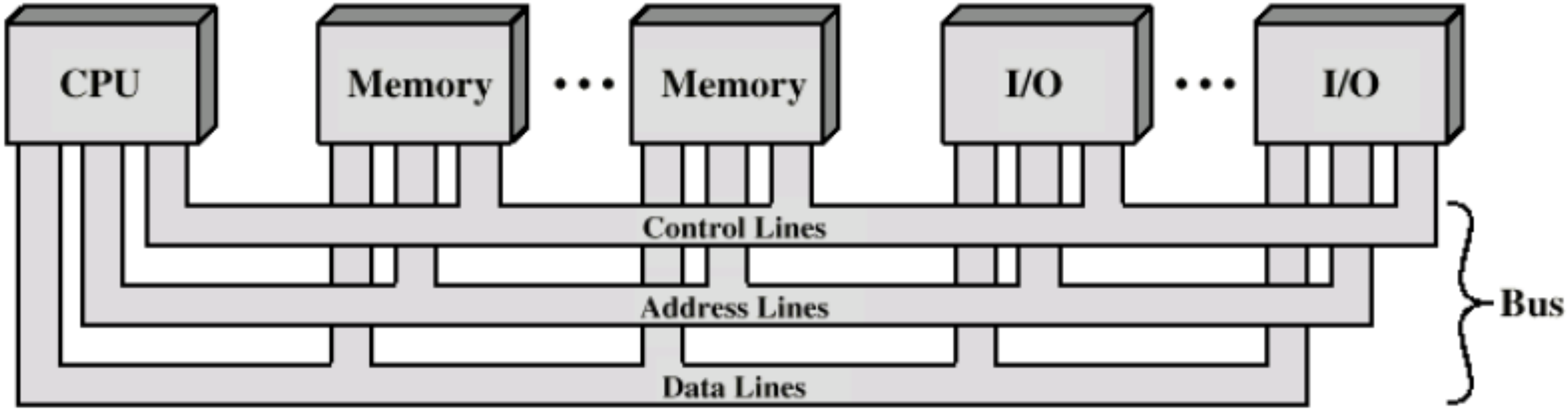
```
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#define PORT1 0x3F8  /* Defines Serial Port Base Address (COM1 */
void main(void){
    unsigned char c = 0;
    unsigned char chrctr = 0;
    /*int exit = 1; */
    outportb(PORT1 + 1, 0); /* Turn off interrupts */
    /* PORT1 Communication Settings */
    outportb(PORT1 + 3, 0x80); /* Set DLAB ON */
    outportb(PORT1 + 0, 0x0C); /* Set the baud rate to 9600 */
    outportb(PORT1 + 1, 0x00); /* Set Baud - Divisor latch HIGH */
    outportb(PORT1 + 3, 0x03); /* 8 bits, no parity, 1 stop */
    outportb(PORT1 + 2, 0xC7); /* FIFO Control Register */
    outportb(PORT1 + 4, 0x0B); /* Turn on DTR, RTS, and OUT2) */
    printf("Waiting on transmission from source.\nPress ESC to quit.\n");
    while(chrctr != 27){      /* Execute the loop if ESC has been hit */
    c = inportb(PORT1 + 5);
    if (c & 0x01){
        chrctr = inportb(PORT1);
        printf("%d",chrctr);
        }
    if (kbhit()){
        chrctr = getch();
        outportb(PORT1, chrctr);
        }
    }
    }"
```

Address

Digital data in computer

A computer has three bus

- Data bus:
- Address bus:
- Control bus:

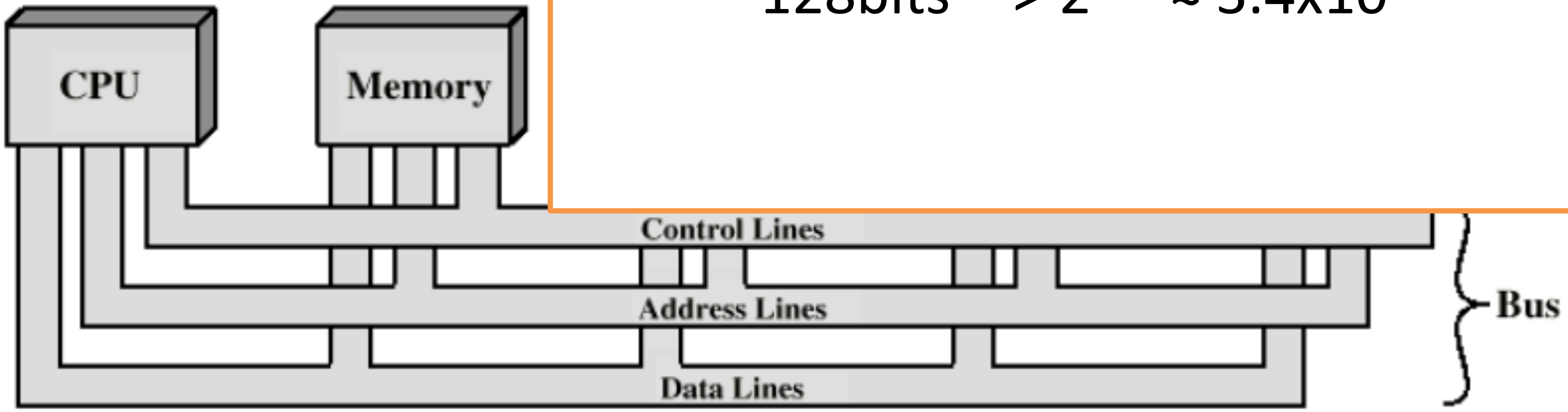


Digital

A computer ha

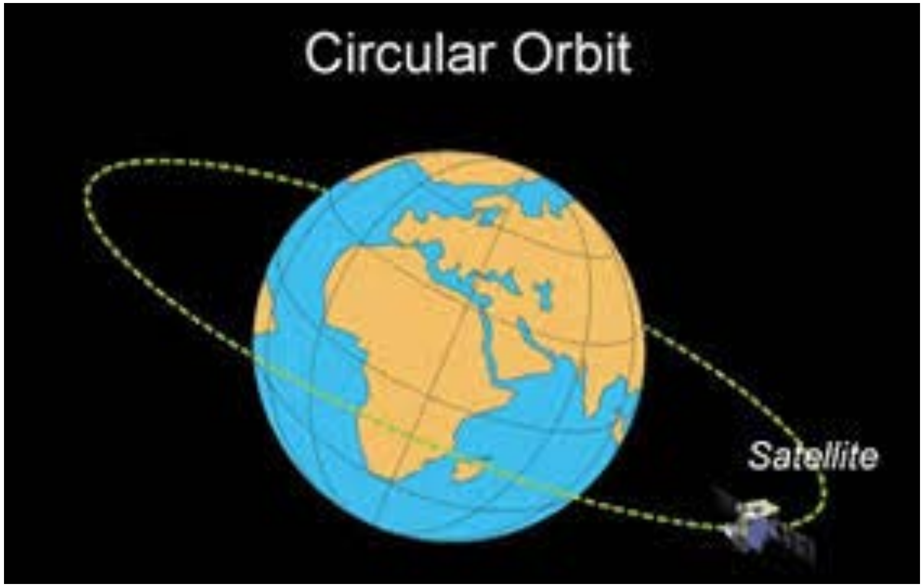
- Data bus:
- Address bus:
- Control bus:

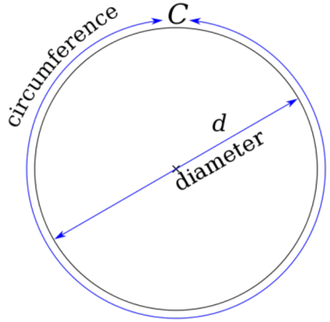
- Data bus carries data or information from CPU/a device to another device.
- Size of data bus
  - 8bits -> 2^8 = 256 levels
  - 16bits -> 2^{16} = 65,536 ≈64K
  - 32bits -> 2^{32} ≈ 4G
  - 64bits -> 2^{64} ≈ 1.8x10^9
  - 128bits -> 2^{128} ≈ 3.4x10^{38}



Why does the computer development from 8 bit to 128 bit?

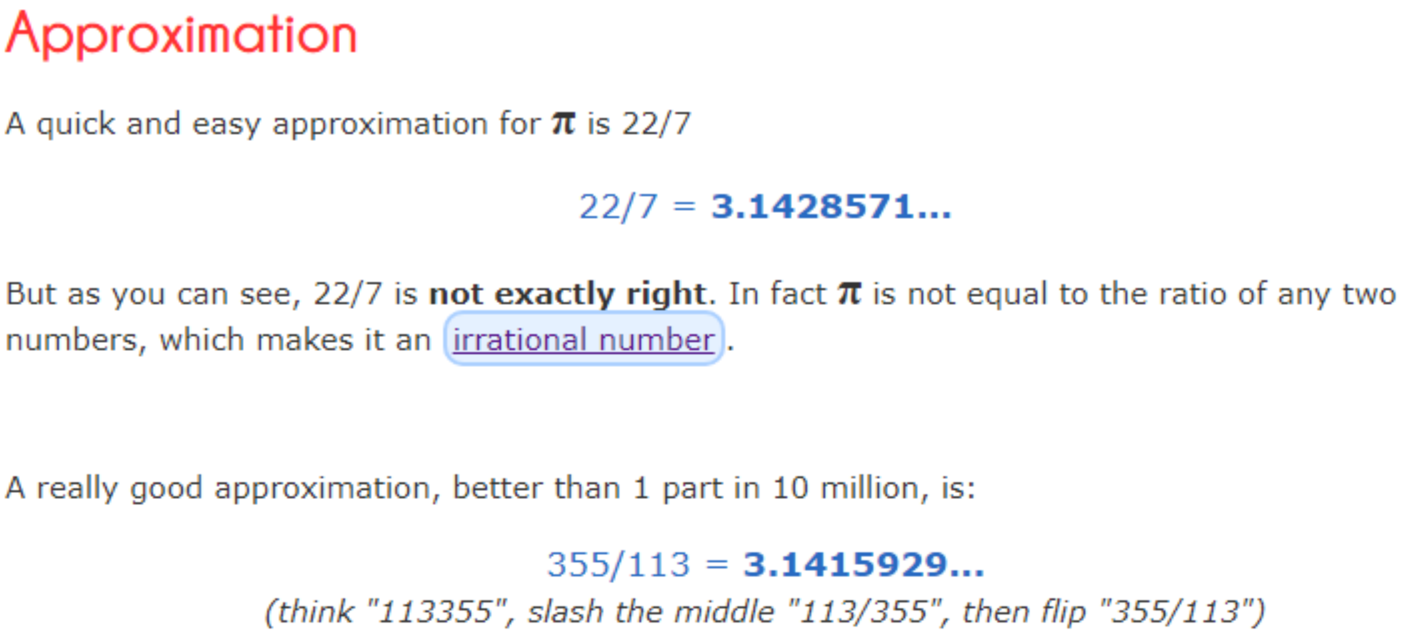
- Calculation circumference between two objects





$C = 2\pi r$

Activity 2.4 Proof Float and Double keep PI



Approximation

A quick and easy approximation for **π** is 22/7

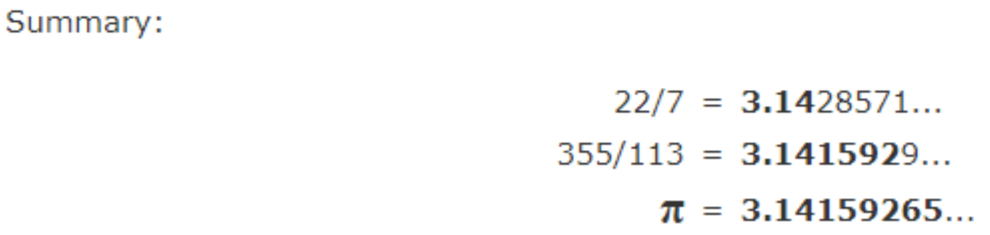
22/7 = **3.1428571...**

But as you can see, 22/7 is **not exactly right**. In fact **π** is not equal to the ratio of any two numbers, which makes it an irrational number.

A really good approximation, better than 1 part in 10 million, is:

355/113 = **3.1415929...**

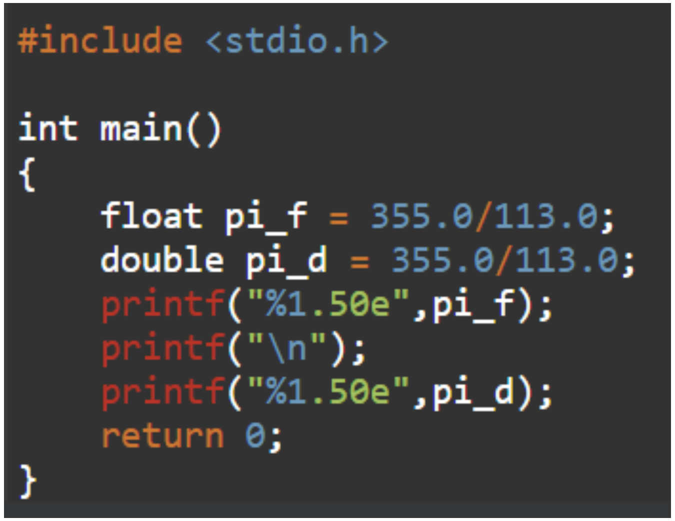
(think "113355", slash the middle "113/355", then flip "355/113")



Summary:

22/7 = **3.14**28571...
355/113 = **3.141592**9...
π = **3.14159265...**

https://www.mathsisfun.com/numbers/pi.html



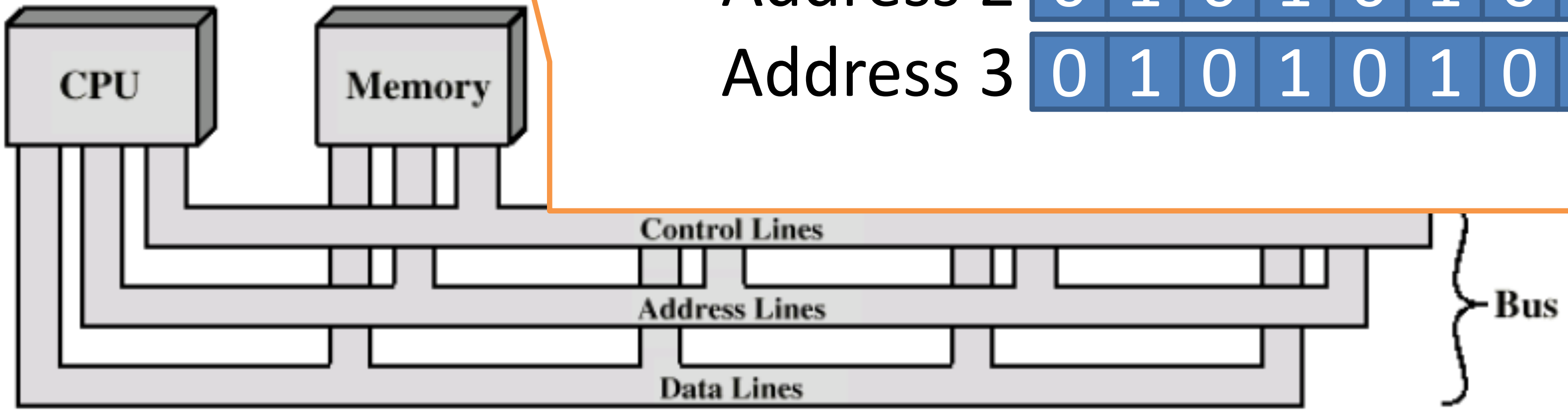
```
#include <stdio.h>

int main()
{
    float pi_f = 355.0/113.0;
    double pi_d = 355.0/113.0;
    printf("%1.50e",pi_f);
    printf("\n");
    printf("%1.50e",pi_d);
    return 0;
}
```

A computer has

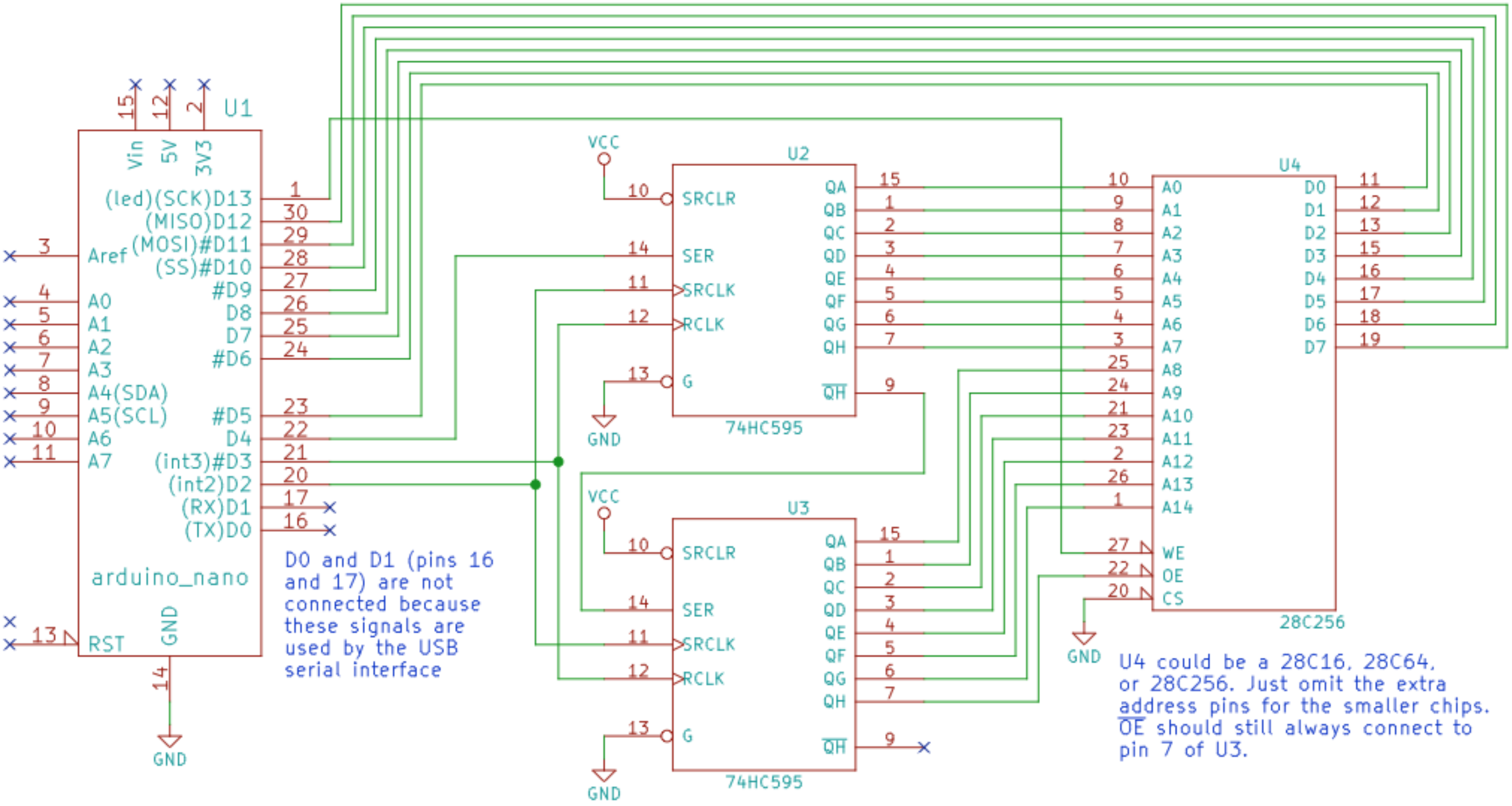
- Data bus:
- Address bus:
- Control bus:

- Address bus relates to position of devices.
- For example memory 4Byte at 8bit has

Address 0	0	1	0	1	0	1	0	1
Address 1	0	1	0	1	0	1	0	1
Address 2	0	1	0	1	0	1	0	1
Address 3	0	1	0	1	0	1	0	1



Activity 2.5, The circuit diagram shown below has 4 ICs, given you consider, which is the chip doing as the memory.



DO and D1 (pins 16 and 17) are not connected because these signals are used by the USB serial interface

U4 could be a 28C16, 28C64, or 28C256. Just omit the extra address pins for the smaller chips. $\overline{\text{OE}}$ should still always connect to pin 7 of U3.

Activity 2.6, The memory IC has the pin name as the list below, given you write the full name of the pin.

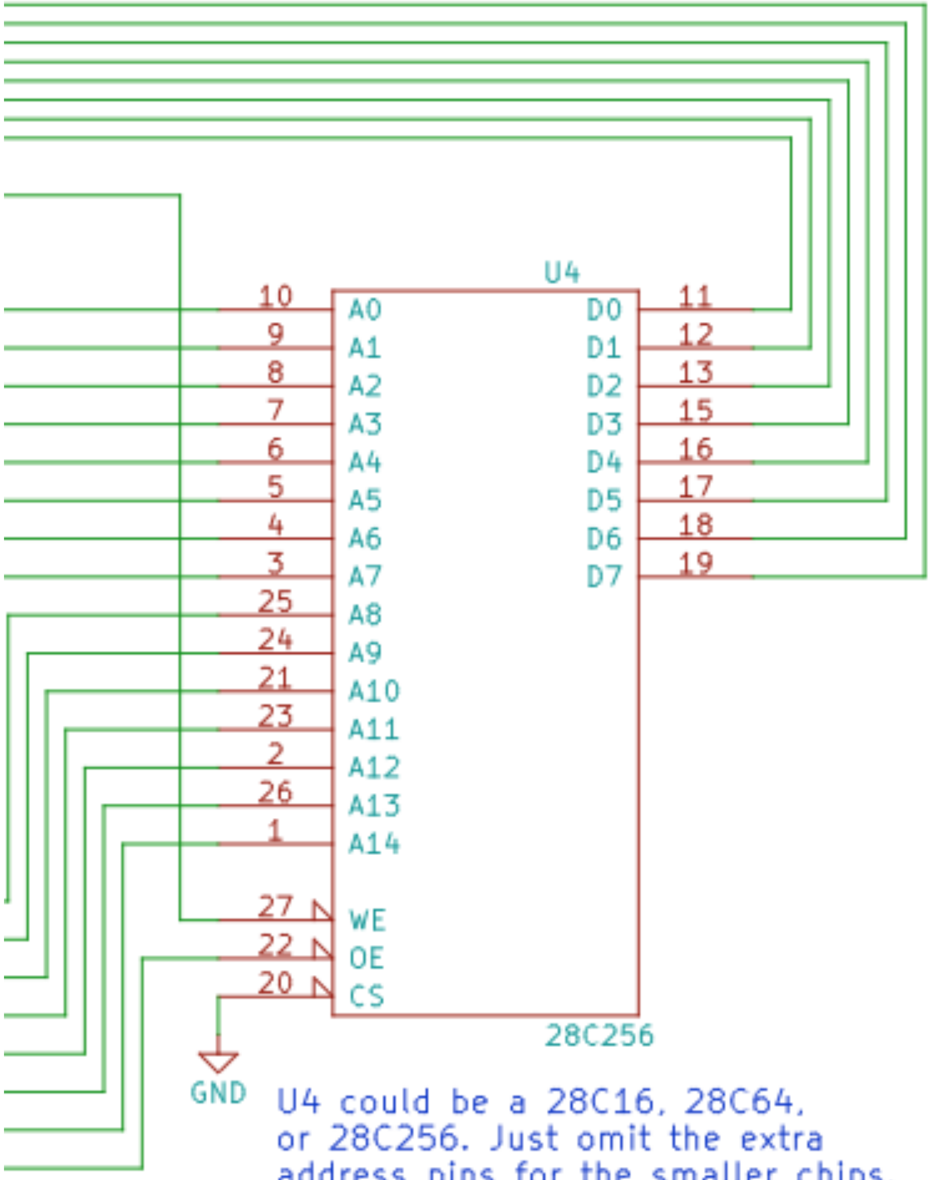
1) An_______________

2) Dn_______________

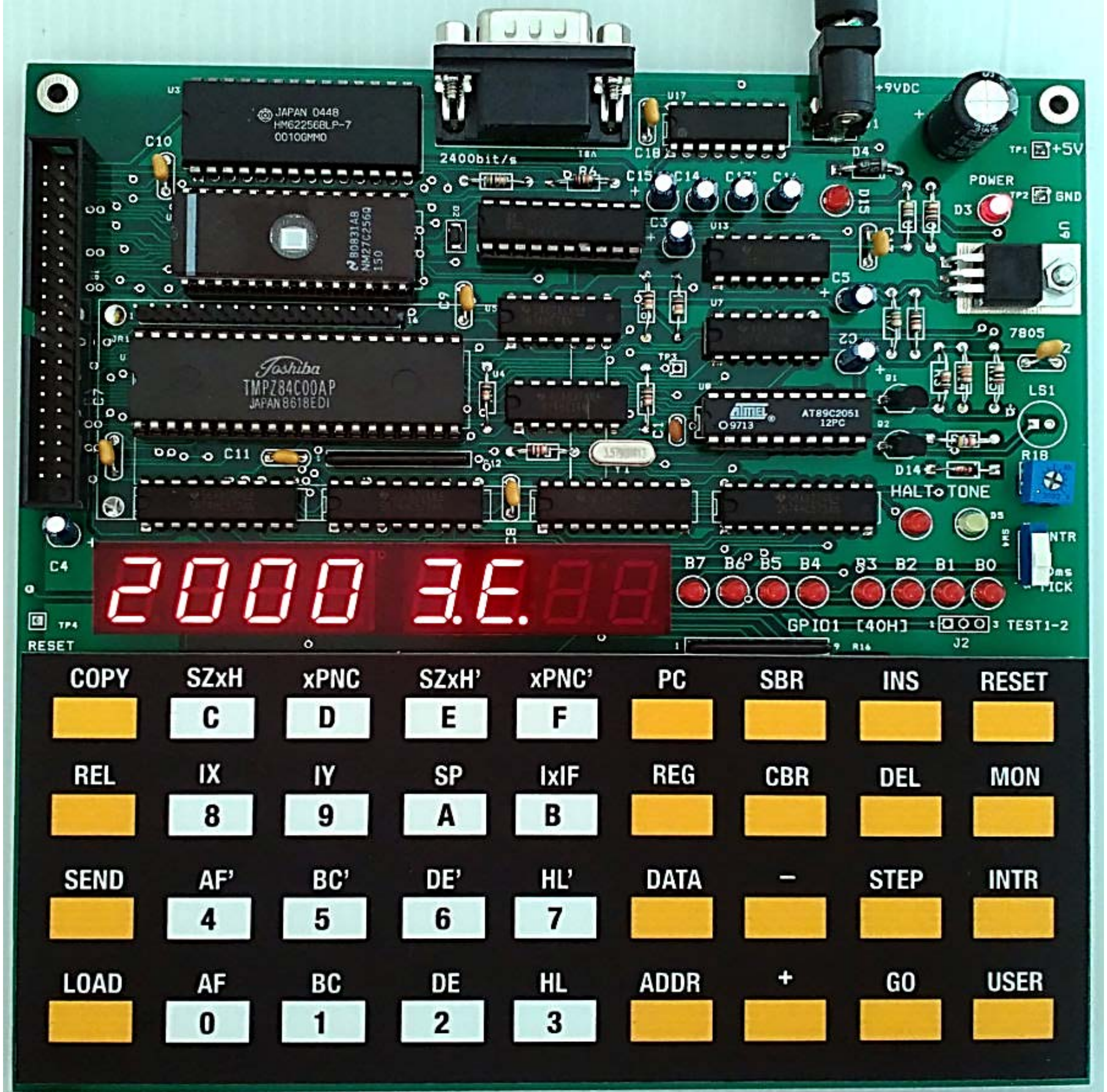
3) WE ______________

4) OE _______________

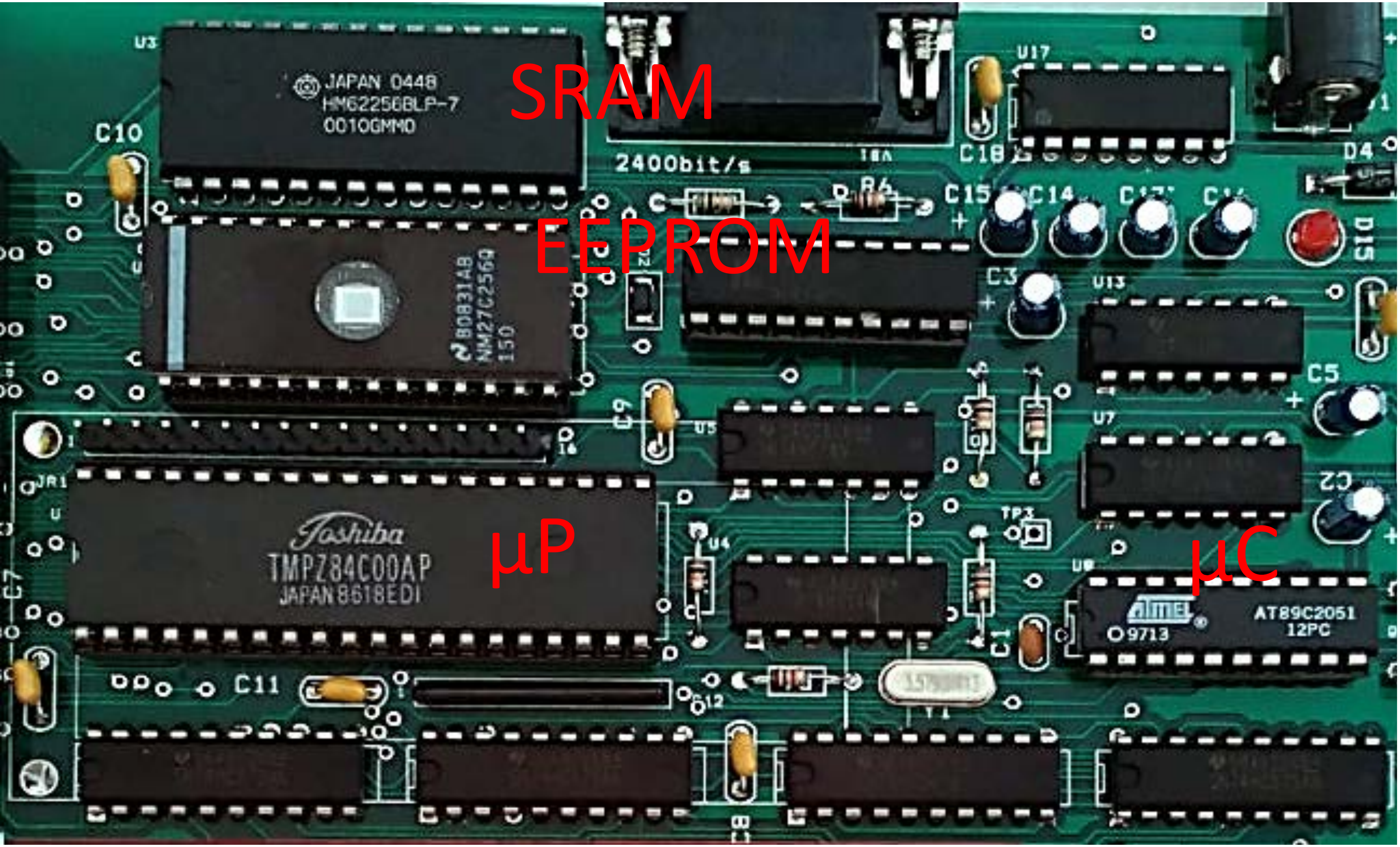
5) CS________________



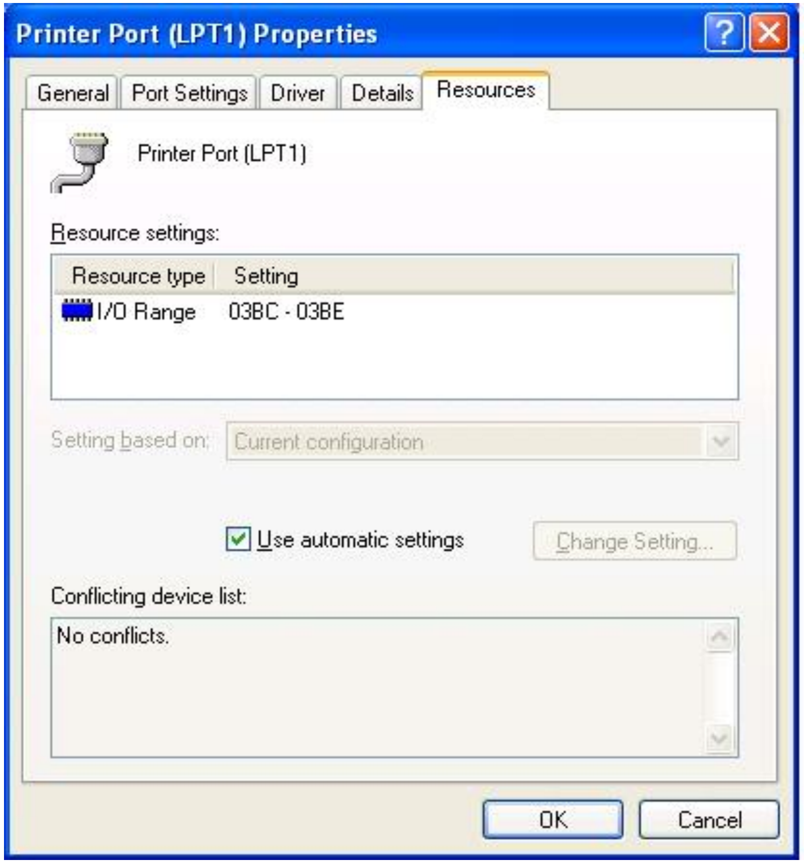
U4 could be a 28C16, 28C64,
or 28C256. Just omit the extra
address pins for the smaller chips.
$\overline{\text{OE}}$ should still always connect to
pin 7 of U3.



2400bit/s
+9VDC
+5V
GND
POWER
7805
LS1
HALT
TONE
INTR
Toshiba
TMPZ84C00AP
HM62256BLP-7
NM27C256Q
AT89C2051
2000 3.E.
B7 B6 B5 B4 B3 B2 B1 B0
GPIO1 [40H]
TEST1-2
RESET
COPY
SZxH
C
xPNC
D
SZxH'
E
xPNC'
F
PC
SBR
INS
RESET
REL
IX
8
IY
9
SP
A
IxIF
B
REG
CBR
DEL
MON
SEND
AF'
4
BC'
5
DE'
6
HL'
7
DATA
–
STEP
INTR
LOAD
AF
0
BC
1
DE
2
HL
3
ADDR
+
GO
USER


SRAM
EEPROM
µP
µC
JAPAN 0448
HM62256BLP-7
0010GMM0
2400bit/s
80831AB
NM27C256Q
150
Toshiba
TMPZ84C00AP
JAPAN 8618EDI
ATMEL
9713
AT89C2051
12PC

Example printer port address in Window XP



Printer Port (LPT1) Properties

General | Port Settings | Driver | Details | Resources

Printer Port (LPT1)

Resource settings:

Resource type	Setting
I/O Range	03BC - 03BE

Setting based on: Current configuration

☑ Use automatic settings

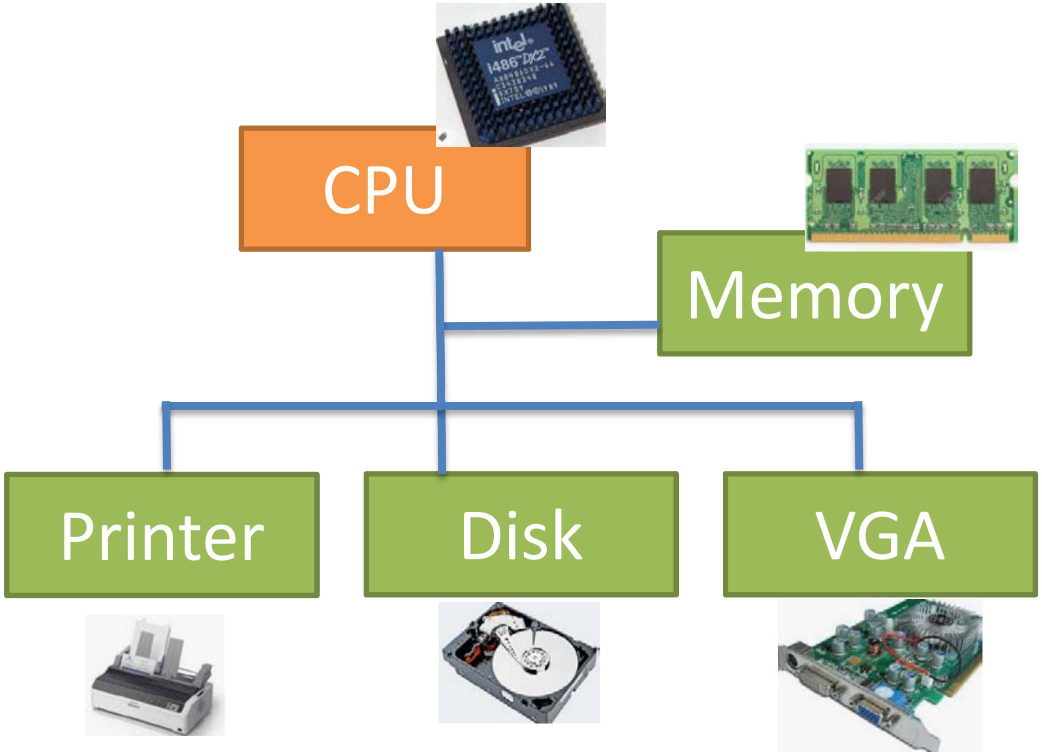
Change Setting...

Conflicting device list:

No conflicts.

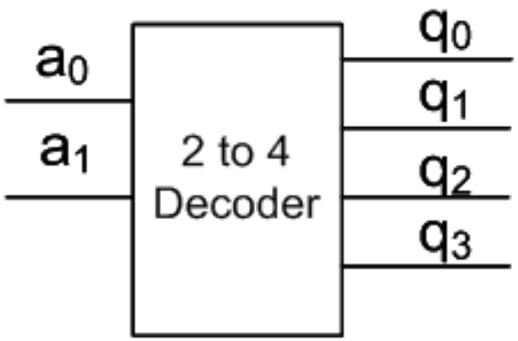
OK

Cancel

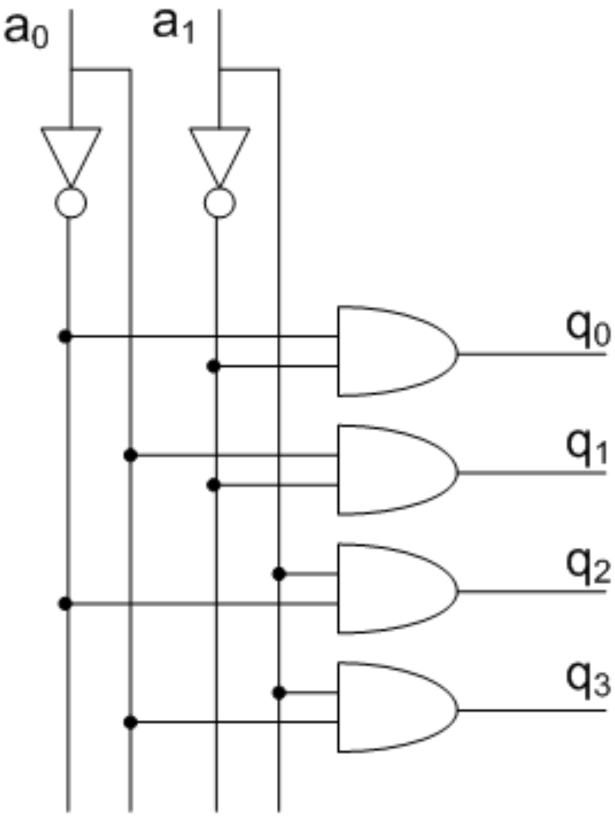


DECODER CIRCUIT

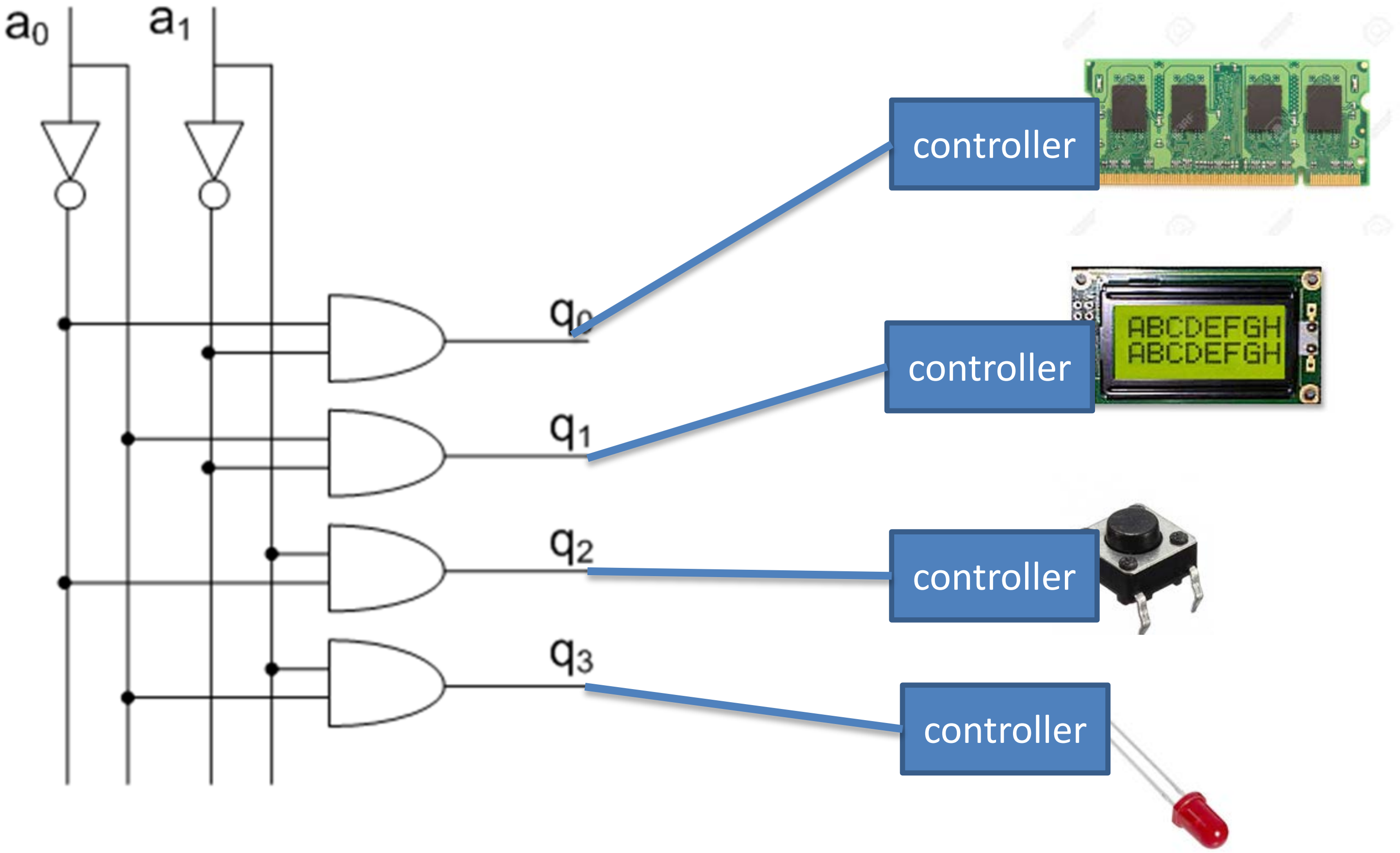
Decoder circuit



a0	**a1**	**q0**	**q1**	**q2**	**q3**
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



Decoder circuit selects devices



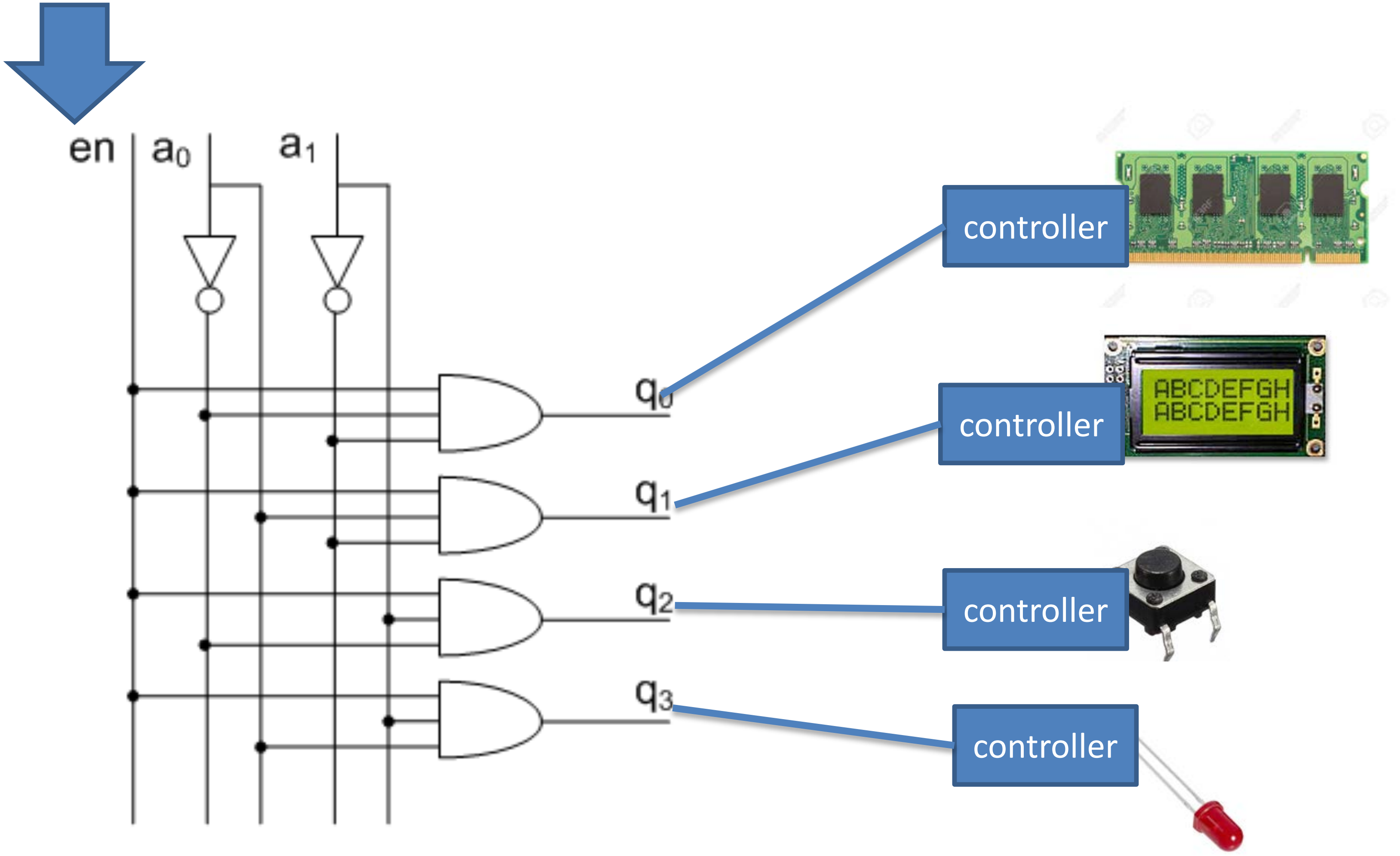
controller

controller

controller

controller

Decoder circuit with enable



controller

controller

controller

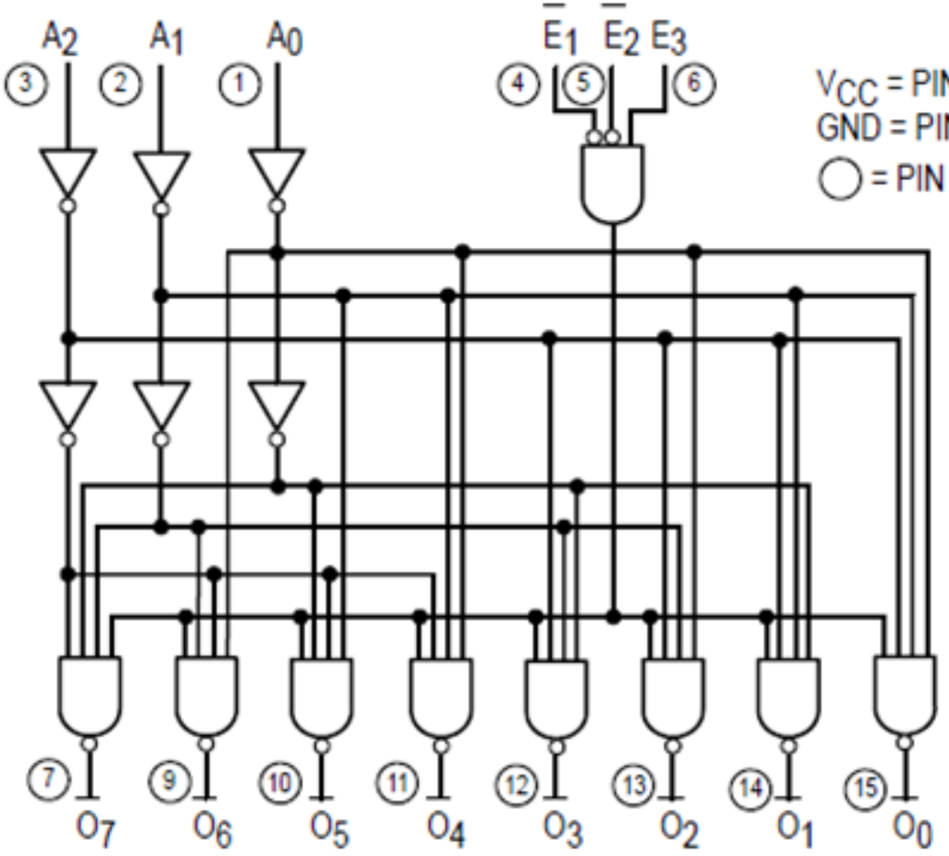
controller

MOTOROLA

1-OF-8 DECODER/ DEMULTIPLEXER

SN54/74LS138

LOGIC DIAGRAM



V_{CC} = PIN 16
GND = PIN 8
○ = PIN NUMBERS

TRUTH TABLE

INPUTS						OUTPUTS							
$\overline{E}_1$	$\overline{E}_2$	E_3	A_0	A_1	A_2	$\overline{O}_0$	$\overline{O}_1$	$\overline{O}_2$	$\overline{O}_3$	$\overline{O}_4$	$\overline{O}_5$	$\overline{O}_6$	$\overline{O}_7$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

Extended address

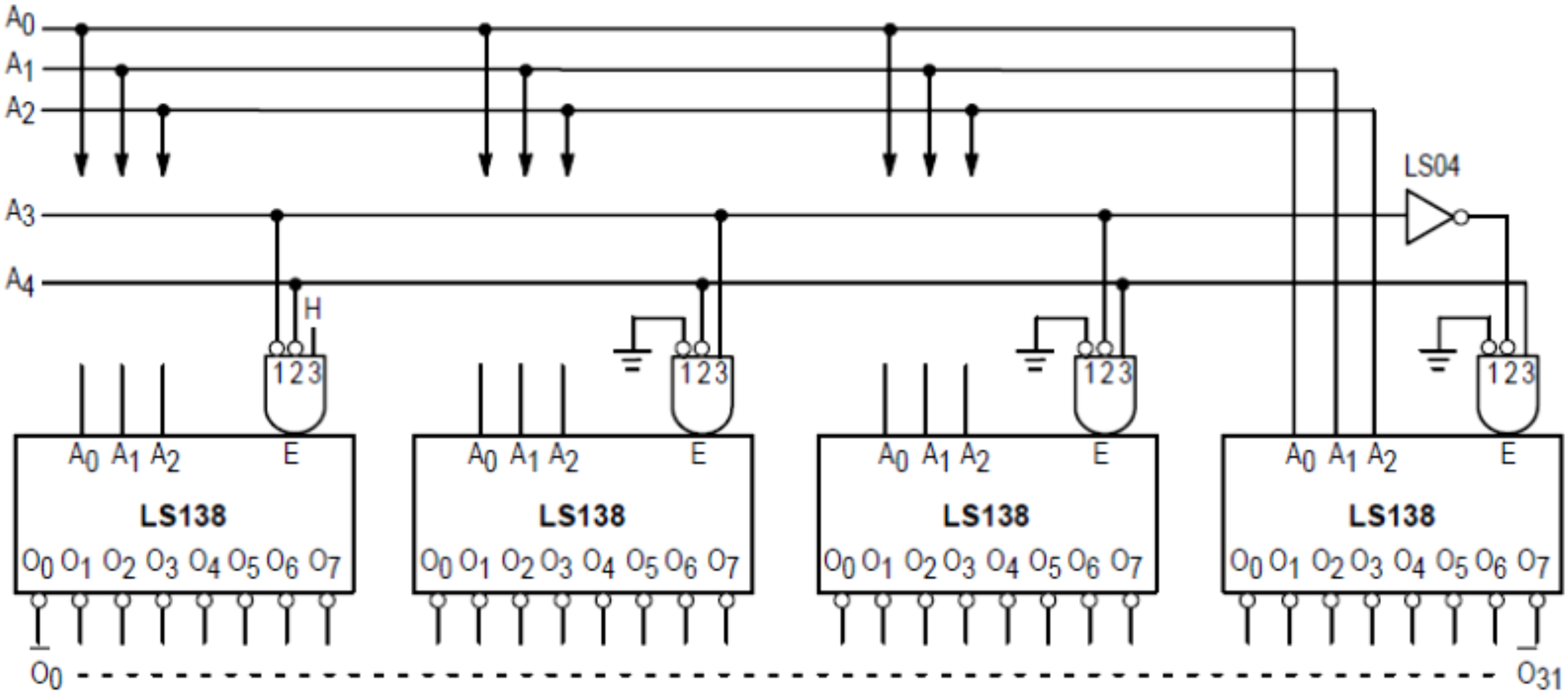
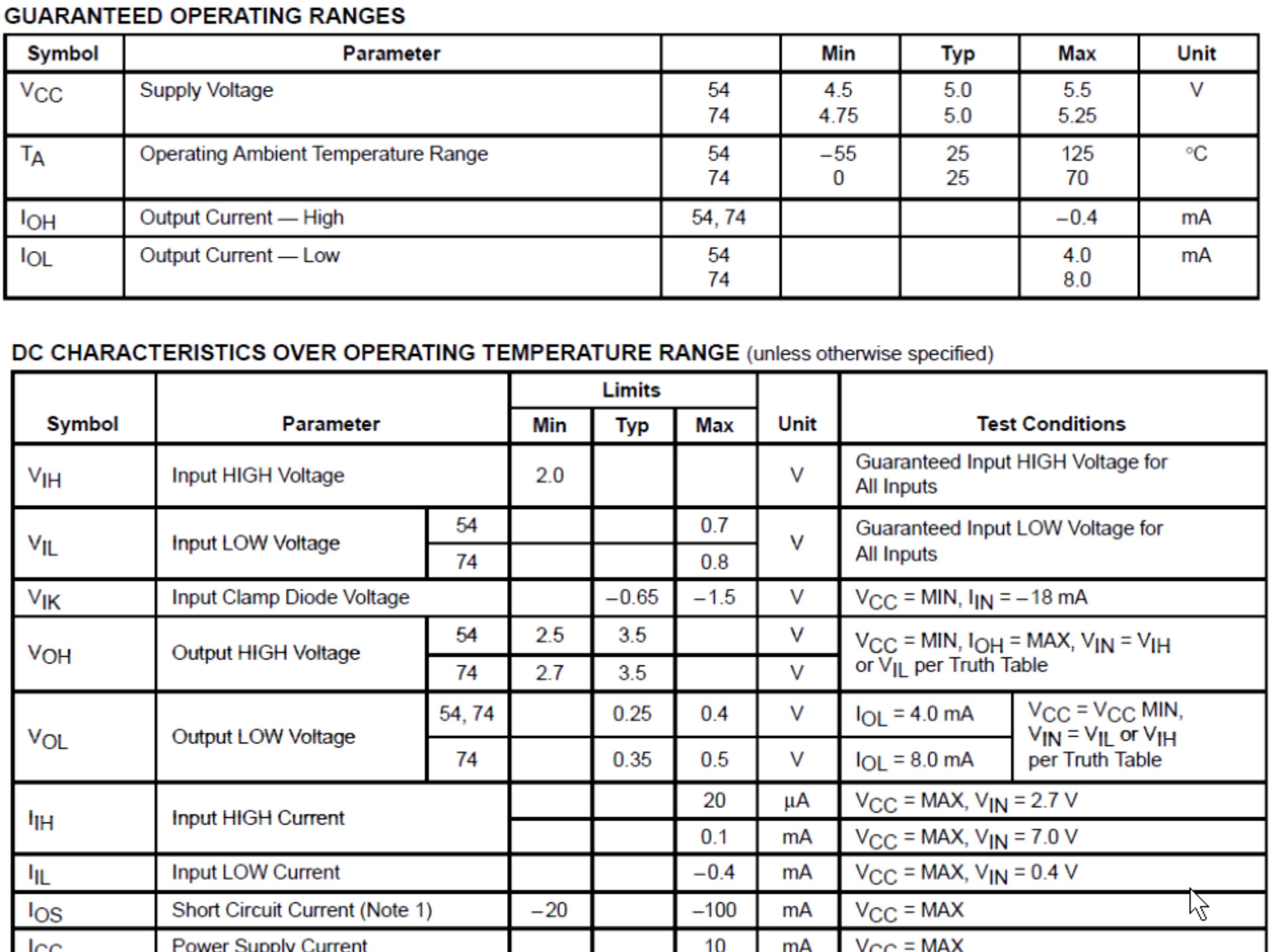


Figure a



GUARANTEED OPERATING RANGES

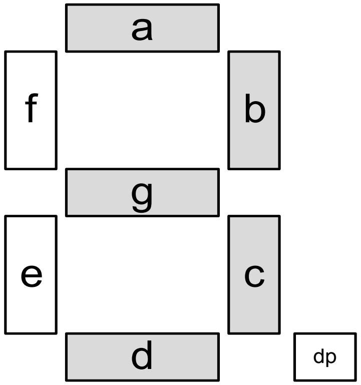
Symbol	Parameter		Min	Typ	Max	Unit
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	–55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54, 74			–0.4	mA
I_{OL}	Output Current — Low	54 74			4.0 8.0	mA

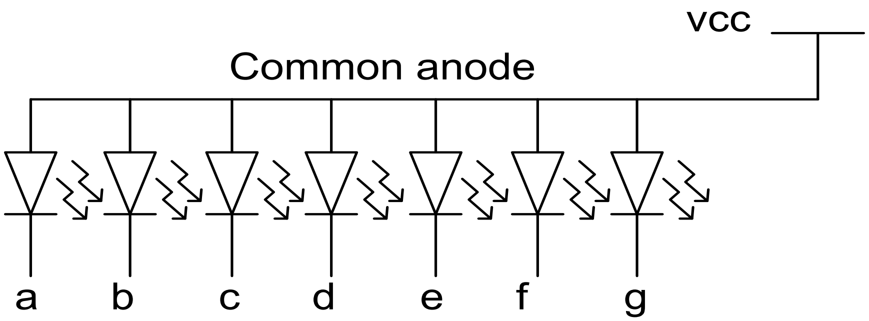
DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

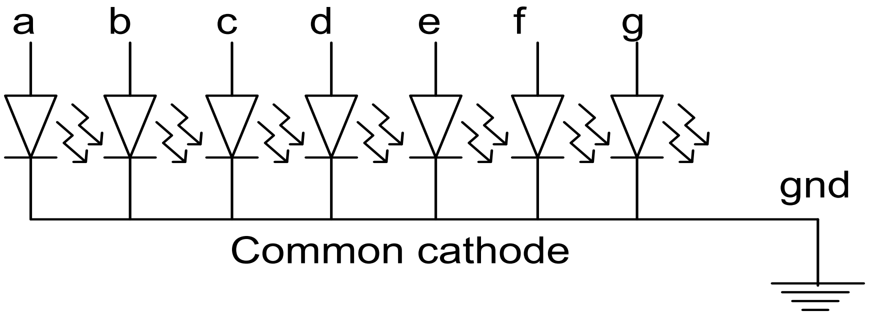
Symbol	Parameter		Limits			Unit	Test Conditions	
			Min	Typ	Max			
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
V_{IK}	Input Clamp Diode Voltage			–0.65	–1.5	V	V_{CC} = MIN, I_{IN} = –18 mA	
V_{OH}	Output HIGH Voltage	54	2.5	3.5		V	V_{CC} = MIN, I_{OH} = MAX, V_{IN} = V_{IH} or V_{IL} per Truth Table	
		74	2.7	3.5		V		
V_{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I_{OL} = 4.0 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 8.0 mA	
I_{IH}	Input HIGH Current				20	µA	V_{CC} = MAX, V_{IN} = 2.7 V	
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
I_{IL}	Input LOW Current				–0.4	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{OS}	Short Circuit Current (Note 1)		–20		–100	mA	V_{CC} = MAX	
I_{CC}	Power Supply Current				10	mA	V_{CC} = MAX	

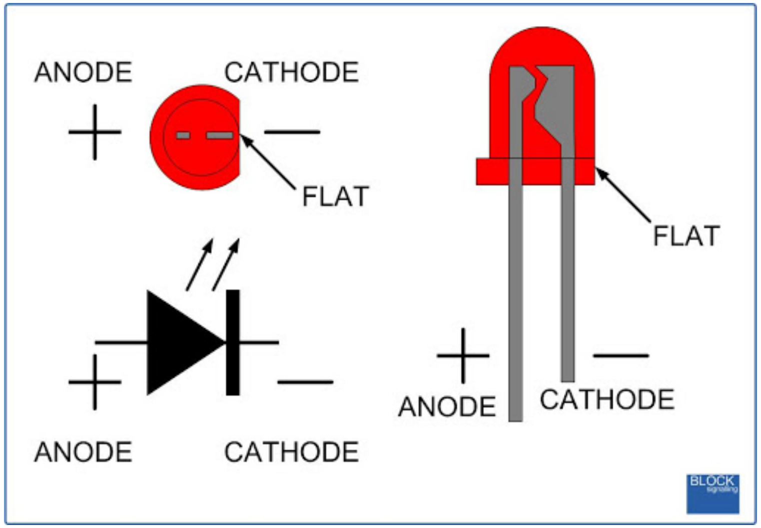
LED SEVEN SEGMENT
COMMON ANODE AND CATHODE

Activity: 2.7 Plot your name on 7-segment

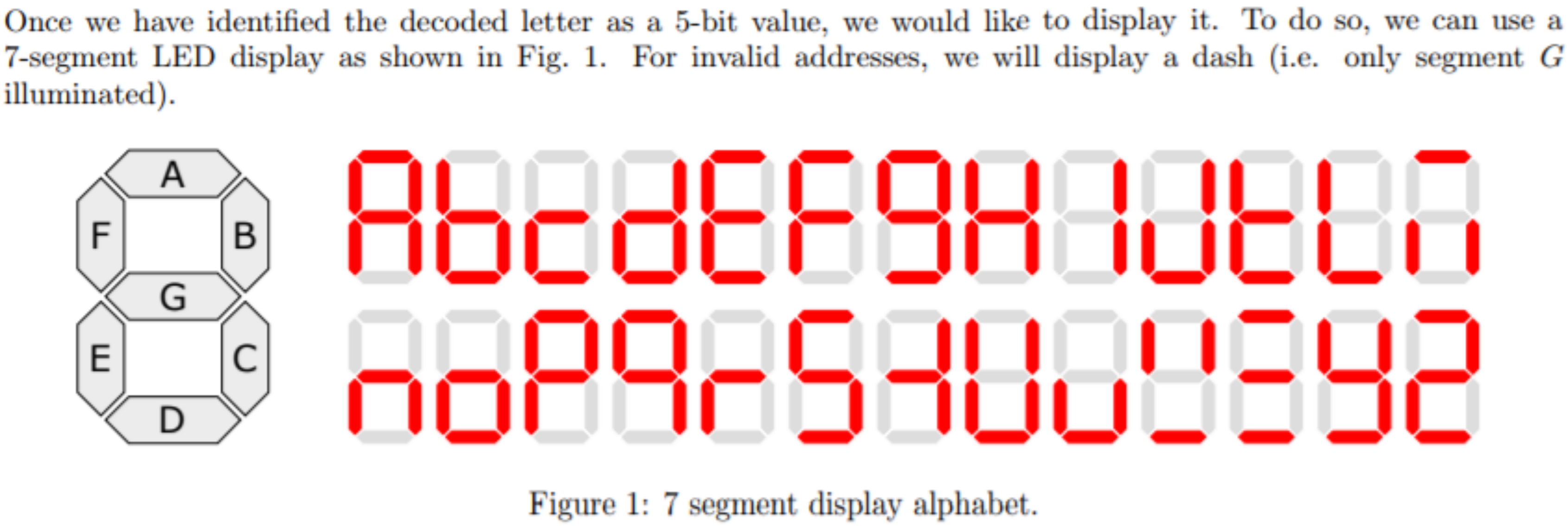








Alphabet present on 7-Segment



Once we have identified the decoded letter as a 5-bit value, we would like to display it. To do so, we can use a 7-segment LED display as shown in Fig. 1. For invalid addresses, we will display a dash (i.e. only segment *G* illuminated).

Figure 1: 7 segment display alphabet.

Summary