

P524: Survey of Instrumentation and Laboratory Techniques Week 1-2

Instructor: Chen-Yu Liu

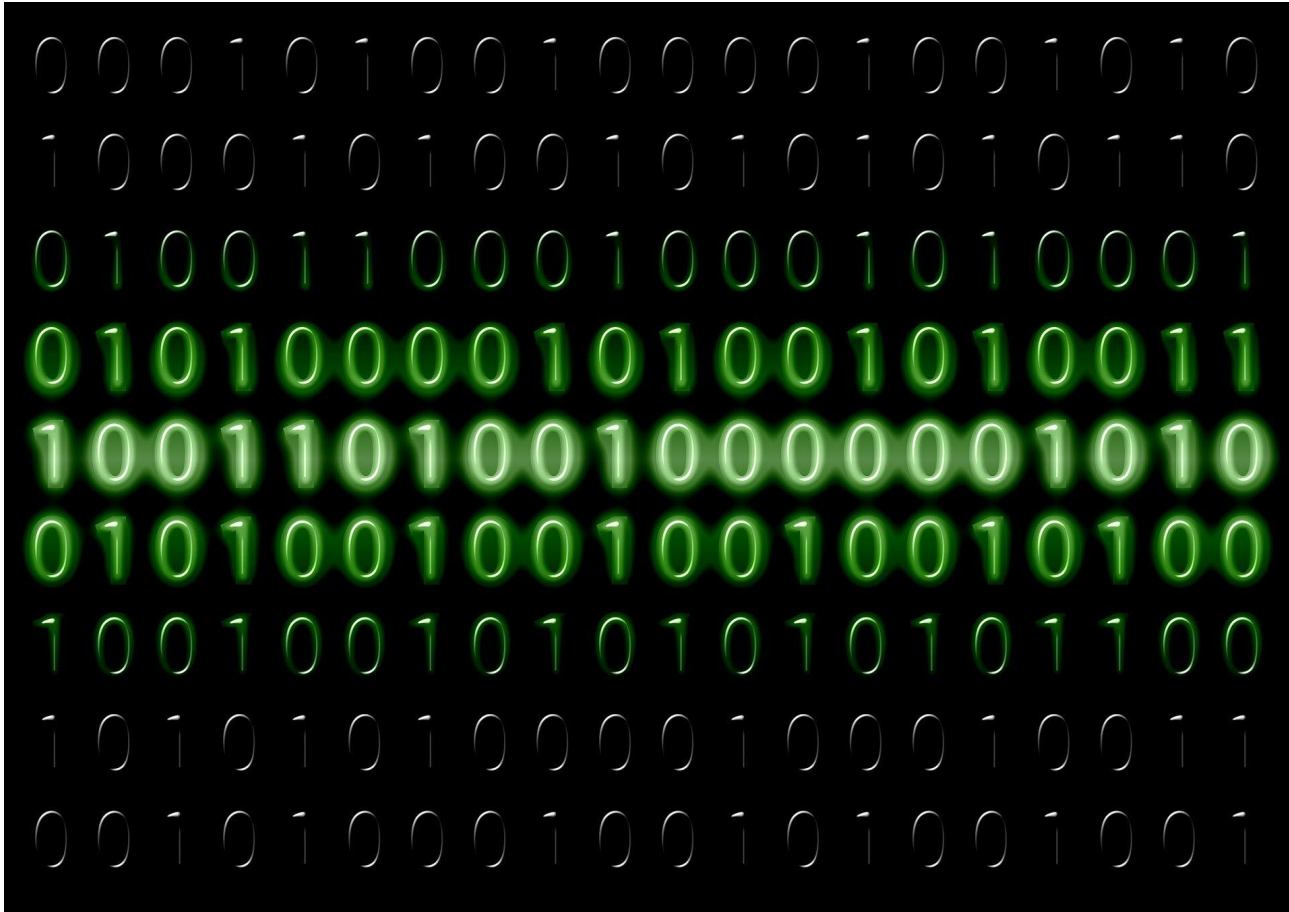
Guest lecturer: Gregory Hallewell, Brian Mercer, Tim Barklow

TA: Chad Lantz

Grader: Garrett Williams

Course designer: George Collins, who will teach P523 (projects)

Digital Computer



- Digital electronics devices store and process bits electronically.
 - A **bit** represents data using 1's and 0's
 - Eight bits is a **byte** – the standard grouping in digital electronics
 - 64-bit machines (your laptop computer), corresponding to eight-byte words.

Number Representation

Decimal	4-Bit Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Decimal representation:

Place: 4th 3rd 2nd 1st

Place value: 10^3 10^2 10^1 10^0

Data: **0** **0** **1** **3**

Value: $0 \times 1000 + 0 \times 100 + 1 \times 10 + 3 \times 1 = 13$

Number Representation

Decimal	4-Bit Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Binary representation:

Place: 4th 3rd 2nd 1st

Place value: 2^3 2^2 2^1 2^0

Data: **1** **1** **0** **1**

Value: $1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 13$

Bit, bytes, and words

- Digital electronics devices store and process bits electronically.
 - A bit represents data using 1's and 0's
 - Eight bits is a byte – the standard grouping in digital electronics
 - Hexadecimal representation of a number by preceding the hex digits with 0x

e.g. Show that the number 63 can be represented as follows:

$0x3F = 0b00111111 = 0d63$

- bytes are grouped together to form words. The number of bytes per word depends on the architecture of a particular memory or processor chip; common values are two, four, and eight bytes per word.
 - Most of your laptops are 64-bit machines, corresponding to eight-byte words.

ASCII table

ASCII stands for American Standard Code for Information Interchange.

On the right is the ASCII character table, including descriptions of the first 32 characters.

ASCII was originally designed for use with teletypes, and so the descriptions are somewhat obscure and their use is frequently not as intended.

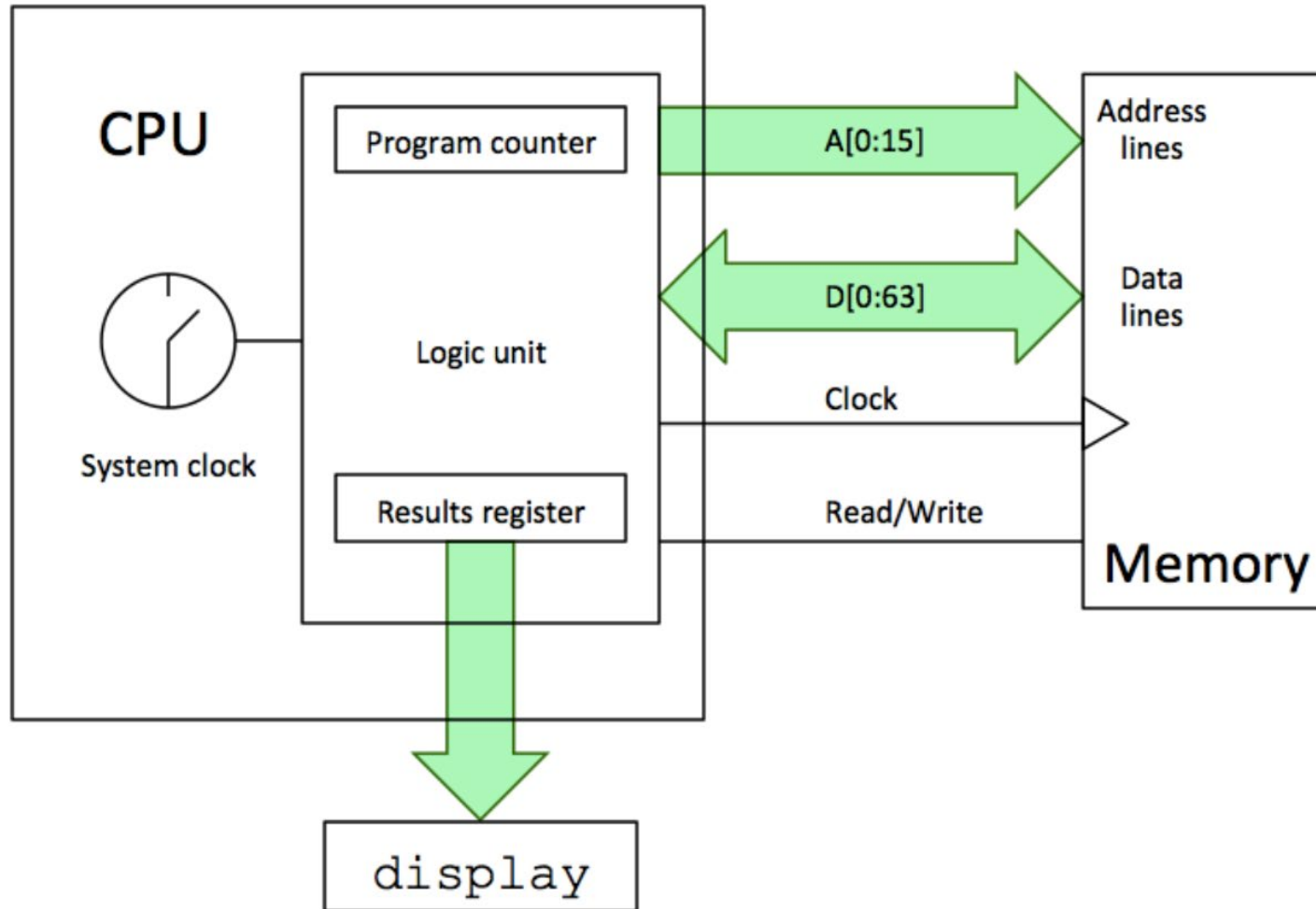
Java actually uses Unicode, which includes ASCII and other characters from languages around the world.

Dec	Char	Dec	Char	Dec	Char	Dec	Char
0	NUL (null)	32	SPACE	64	@	96	`
1	SOH (start of heading)	33	!	65	A	97	a
2	STX (start of text)	34	"	66	B	98	b
3	ETX (end of text)	35	#	67	C	99	c
4	EOT (end of transmission)	36	\$	68	D	100	d
5	ENQ (enquiry)	37	%	69	E	101	e
6	ACK (acknowledge)	38	&	70	F	102	f
7	BEL (bell)	39	'	71	G	103	g
8	BS (backspace)	40	(72	H	104	h
9	TAB (horizontal tab)	41)	73	I	105	i
10	LF (NL line feed, new line)	42	*	74	J	106	j
11	VT (vertical tab)	43	+	75	K	107	k
12	FF (NP form feed, new page)	44	,	76	L	108	l
13	CR (carriage return)	45	-	77	M	109	m
14	SO (shift out)	46	.	78	N	110	n
15	SI (shift in)	47	/	79	O	111	o
16	DLE (data link escape)	48	0	80	P	112	p
17	DC1 (device control 1)	49	1	81	Q	113	q
18	DC2 (device control 2)	50	2	82	R	114	r
19	DC3 (device control 3)	51	3	83	S	115	s
20	DC4 (device control 4)	52	4	84	T	116	t
21	NAK (negative acknowledge)	53	5	85	U	117	u
22	SYN (synchronous idle)	54	6	86	V	118	v
23	ETB (end of trans. block)	55	7	87	W	119	w
24	CAN (cancel)	56	8	88	X	120	x
25	EM (end of medium)	57	9	89	Y	121	y
26	SUB (substitute)	58	:	90	Z	122	z
27	ESC (escape)	59	;	91	[123	{
28	FS (file separator)	60	<	92	\	124	
29	GS (group separator)	61	=	93]	125	}
30	RS (record separator)	62	>	94	^	126	~
31	US (unit separator)	63	?	95	_	127	DEL

A floating point number (64-bit, or 8 byte)

what	which bit(s)	range of values
sign	63 (1 bit)	[0, 1]
exponent	52 – 62 (11 bits)	[-1022, 1023]
significand	0 – 51 (52 bits)	[0, 4.5035996×10^{15}]

A simple model of a computer



- The model of a computer includes a simple CPU, including its communication lines and one register, a part of the CPU that communicates with the outside world.
- Each time the system clock “ticks,” the Logic unit fetches, then executes an instruction from the memory address specified in the Program counter.
- Instructions for our toy computer might contain three fields: *an operation code* (op code) and *a pair of addresses a_1 and a_2* . To add the contents of a_1 to the contents of a_2 , storing the result in a_2 , we’d have the operating system load an instruction into memory with the appropriate op code and address values.