Unit I
(Data representation & Basic data types)

Lecture Slide

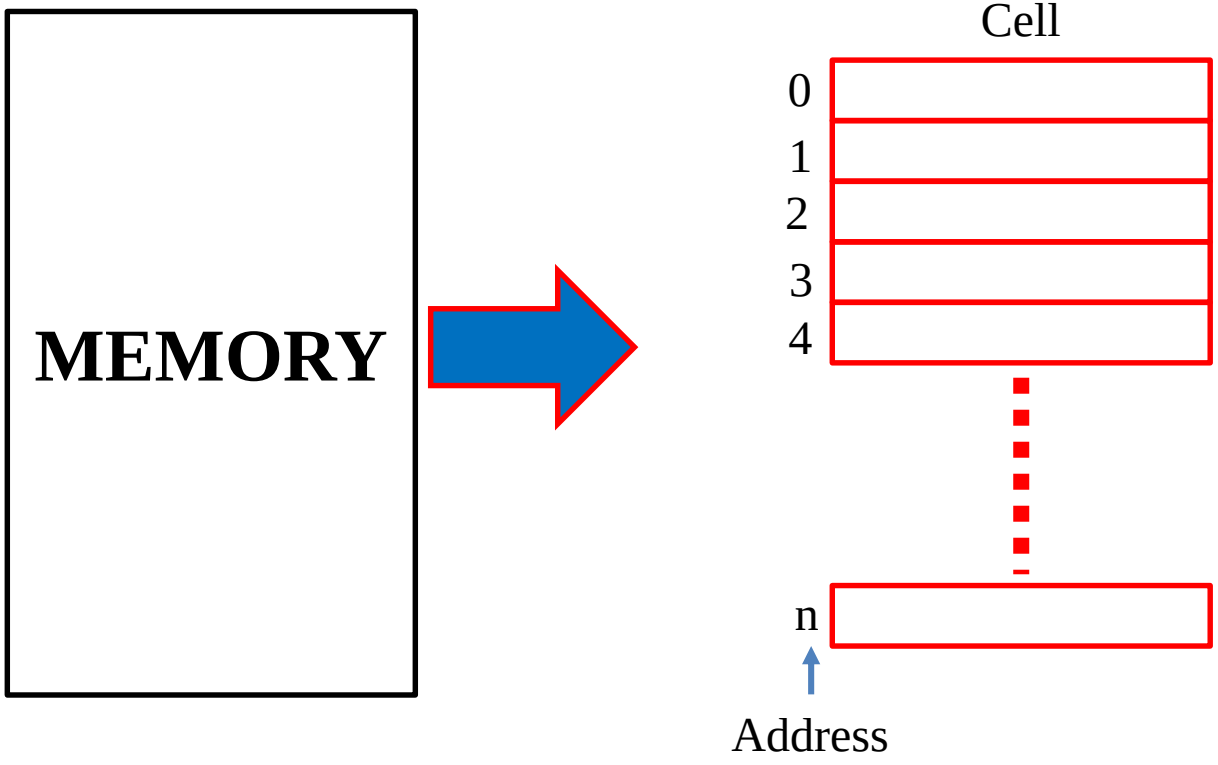
AS2023

Objectives

By the end of this session, students will be able to:

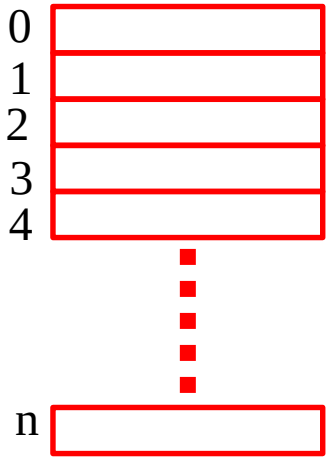
- Demonstrate memory segments
- Explain how data are organized in the memory (Endian)
- Explain basic data types (Integer, Float and Character)
- Elaborate IEEE 754 floating point representations
- Convert the real numbers into single and double precision IEEE 754 FP
- Explain character representation into computer (ASCII code and Unicode)

Memory



Endian

- Each cell stores typically **8 bits** or **1 byte**
- Data can be of more than 8 bits
- Consider E2564F52, 32 bit or 4 byte data, how is it stored in the memory?

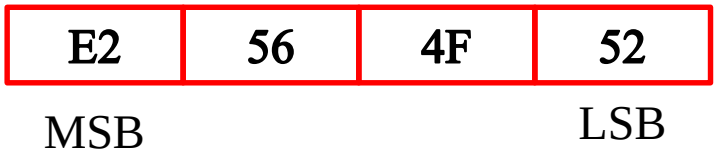


Endian

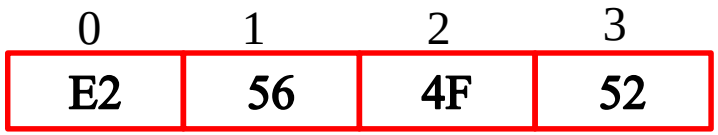
- Endian describes the order in which a sequence of bytes are stored in a memory
- Each data has its MSB and LSB
- Referring to the first byte or value stored in the memory, endianness can be of
  - **Little endian** (LSB at the lowest address)
  - **Big endian** (MSB at the lowest address)

Endian

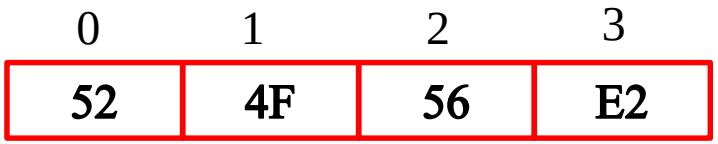
- Given E2564F52 is a 32bit or 4 byte data



- **Big Endian**



- **Little Endian**



Real Numbers

- Union of rational and irrational numbers
- Can be signed either positive or negative
- **Natural numbers**, **decimals** and **fractions** come under this category
- **Integers:** any signed whole number without fractional part
  - **Eg.** -2, 7
- **Floating:** whole numbers with fractional part
  - **Eg.** 2.5, 3.50

Scientific notation

- Floating-point representation uses scientific notation to encode numbers, with a **mantissa** and an **exponent**
  - Decimal number **123.456** represented as

    1.23456 ×10^{2}
  - Hexadecimal number **123.abc** represented as

    1.23abc ×16^{2}
  - Binary number **10100.110** represented as

    1.0100110 ×2^{4}
- Generally, scientific notation can be written as **M** x **base**$^{\text{Exponent}}$

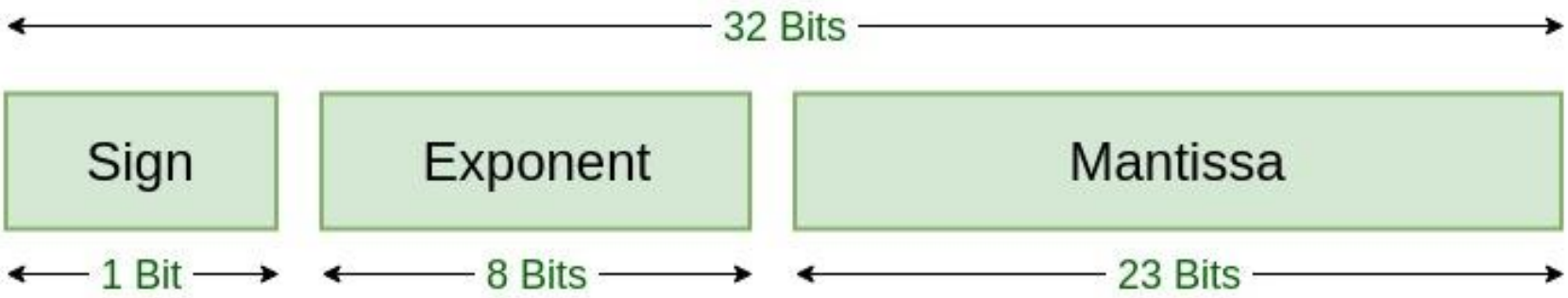
IEEE 754 FPR

- Most common representation today for real numbers
- IEEE 754 has three basic components:
  - Sign
  - Exponent
  - Mantissa
- **Sign:** 0 for positive and 1 for negative
- **Mantissa**
  - The mantissa is part of a number in scientific notation or a floating-point number, consisting of its significant digits.
  - Here we have only 2 digits, i.e. 0 and 1.
  - So a Normalized mantissa is one with only one 1 to the left of the decimal.

IEEE 754 FPR (Cont..)

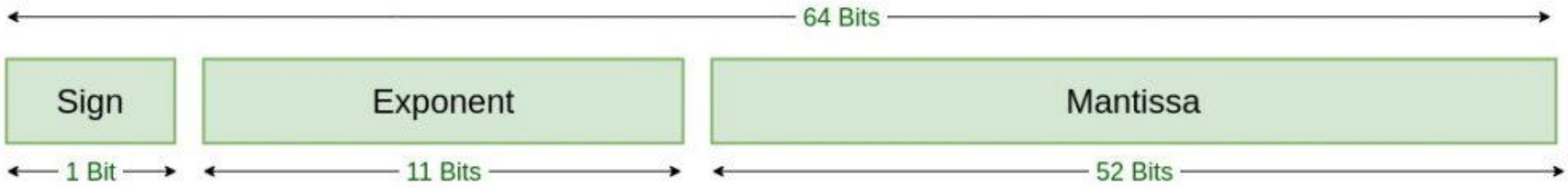
- **Exponent:**
  - The exponent field needs to represent both positive and negative exponents.
  - A **bias** is added to the **actual exponent** in order to get the **stored exponent**.
  - Bias for single precision is **127** and **1023** for double precision
- For example: $(10.10)_2$ is a real number
  - can be represented as 1.010×2^1
  - Here actual exponent is **1**
  - The **stored exponent** is **128** for single precision and **1024** for double precision

IEEE 754 FPR (Cont..)



Single Precision
IEEE 754 Floating-Point Standard

IEEE 754 FPR (Cont..)



Double Precision
IEEE 754 Floating-Point Standard

IEEE 754 FPR (Cont..)

TYPES	SIGN	BIASED EXPONENT	NORMALISED MANTISA	BIAS
Single precision	1(31st bit)	8(30-23)	23(22-0)	127
Double precision	1(63rd bit)	11(62-52)	52(51-0)	1023

Conversion

- Convert **85.125** to single and double precision IEEE 754 floating point representation

Solutions:

1. **Convert 85.125 to binary number.**
   - Since it is a decimal with fractional part, we need to convert separately
   - We get the binary equivalent of 85 as **1010101** using **repeated division method by 2**

Conversion (cont..)

- We compute the binary equivalent of .125 as follows

Product	Integer Part	Fractional Part
0.125 x 2 = 0.25	**0** **MSB**	.25
0.25 x 2 = 0.5	**0**	.5
0.5 x 2 = 1.0	**1** **LSB**	.0

- Therefore, binary equivalent of **.125** is **001**
- And, $(85.125)_{10}$ = $(\mathbf{1010101.001})_{2}$

Conversion (Cont..)

2. **Expressed binary equivalent into normalized scientific notations**

   1010101.001 can be expressed as

   1.010101001 x 2^6

3. From step 2, we get
   1. Sign as **0**, since the number being positive
   2. Actual Exponent as **6**
   3. Mantissa **010101001**

Conversion (Cont..)

4. **A) Expressing in single precision (32bit) floating point representation**
   - **Stored Exponent = bias + Actual Exponent**
   - **127 + 6 = 133**
   - **133** can be represented as **10000101**

- **Answer**

85.125 = **0**10000101**01010100100000000000000**

Conversion (Cont..)

4. **B) Expressing in double precision (64 bit) floating point representation**
   - **Stored Exponent = bias + Actual Exponent**
   - **1023 + 6 = 1029**
   - **1029** can be represented as **10000000101**

0	10000000101	0101010010000000000000000000000000000000000000000000
Sign	Exponent	Mantissa

- **Answer**

85.125 =
0100001010101001000000000000000000000000000000000000000000000

CLASS ACTIVITY

CLASS ACTIVITY

Convert $(-130.375)_{10}$ into IEEE 754 FPR both in single and double precision.

CLASS ACTIVITY

Convert $(747.51)_8$ into IEEE 754 FPR both in single and double precision.

Character Data Type

- The most basic data type in C.
- It stores a single character and requires a single byte of memory in almost all compilers.
- Everything represented by a computer is represented by binary sequences.
- For computer communication, characters have to be encoded into bytes.
- Character representations
  - ASCII Code
  - Unicode

ASCII Code

- American Standard Code for Information Interchange
- It was designed in the early 60's, as a standard character set for computers and electronic devices.
- ASCII is a 7-bit character set containing 128 characters.
- It contains the numbers from 0-9, the upper and lower case English letters from A to Z, and some special characters.
- The character sets used in modern computers, in HTML, and on the Internet, are all based on ASCII.

ASCII Code

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Home Assignment

- Write short notes on Unicode representation.
- Write short history on Endian.
- Write short notes on the following data type:
  - Integer
  - Float
  - String
- Represent the following in 32 and 64 bit IEEE 754 FP
  1. $(-130.375)_{10}$
  2. $(765.45)_{8}$
  3. $(A5.E5)_{16}$

Thank you