COMPUTER SYSTEMS FUNDAMENTALS
(4COSC004W)

Lecture: Week 2. Part 2 of 2

In this video we will cover:

- Ranges of values
  - *Positive Integers*
    - Decimal
    - Binary
    - Hex
  - *Why use Binary?*
  - *Why use Hexadecimal ?*

RANGE OF VALUES

Positive Integers

By the end of this unit, you will:

- Be able to work out the range of values of Positive (Unsigned) Integers in
  - *Decimal*
  - *Binary*
  - *Hexadecimal*
- Appreciate what we use Binary for
  - *and why*
- Appreciate what we use Hexadecimal for
  - *and why*

Denary	Binary				Hexadecimal
0				0	0
1				1	1
2			1	0	2
3			1	1	3
4		1	0	0	4
5		1	0	1	5
6		1	1	0	6
7		1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	A
11	1	0	1	1	B
12	1	1	0	0	C
13	1	1	0	1	D
14	1	1	1	0	E
15	1	1	1	1	F

Denary	Binary	Hexadecimal
0	0 0 0 0	0
1	0 0 0 1	1
2	0 0 1 0	2
3	0 0 1 1	3
4	0 1 0 0	4
5	0 1 0 1	5
6	0 1 1 0	6
7	0 1 1 1	7
8	1 0 0 0	8
9	1 0 0 1	9
10	1 0 1 0	A
11	1 0 1 1	B
12	1 1 0 0	C
13	1 1 0 1	D
14	1 1 1 0	E
15	1 1 1 1	F

Denary – Base 10

- 1 digit
  - *Values: 0 → 9*
    - $0 \rightarrow (10^1 - 1)$
  - 10^1 *values*
- 2 digits
  - *Values: 0 → 99*
    - $0 \rightarrow (10^2 - 1)$
  - 10^2 *values*
- *n* digits
  - *Values: 0 →* $(10^n - 1)$
  - 10^n *values*

Binary – Base 2

- 1 Bit
  - *Values: 0 → 1*
    - $0 \rightarrow (2^1 - 1)$
  - 2^1 *values*
- 2 Bits
  - *Values: 0 → 3*
    - $0 \rightarrow (2^2 - 1)$
  - 2^2 *values*
- *n* Bits
  - *Values: 0 →* $(2^n - 1)$
  - 2^n *values*

Hexadecimal – Base 16

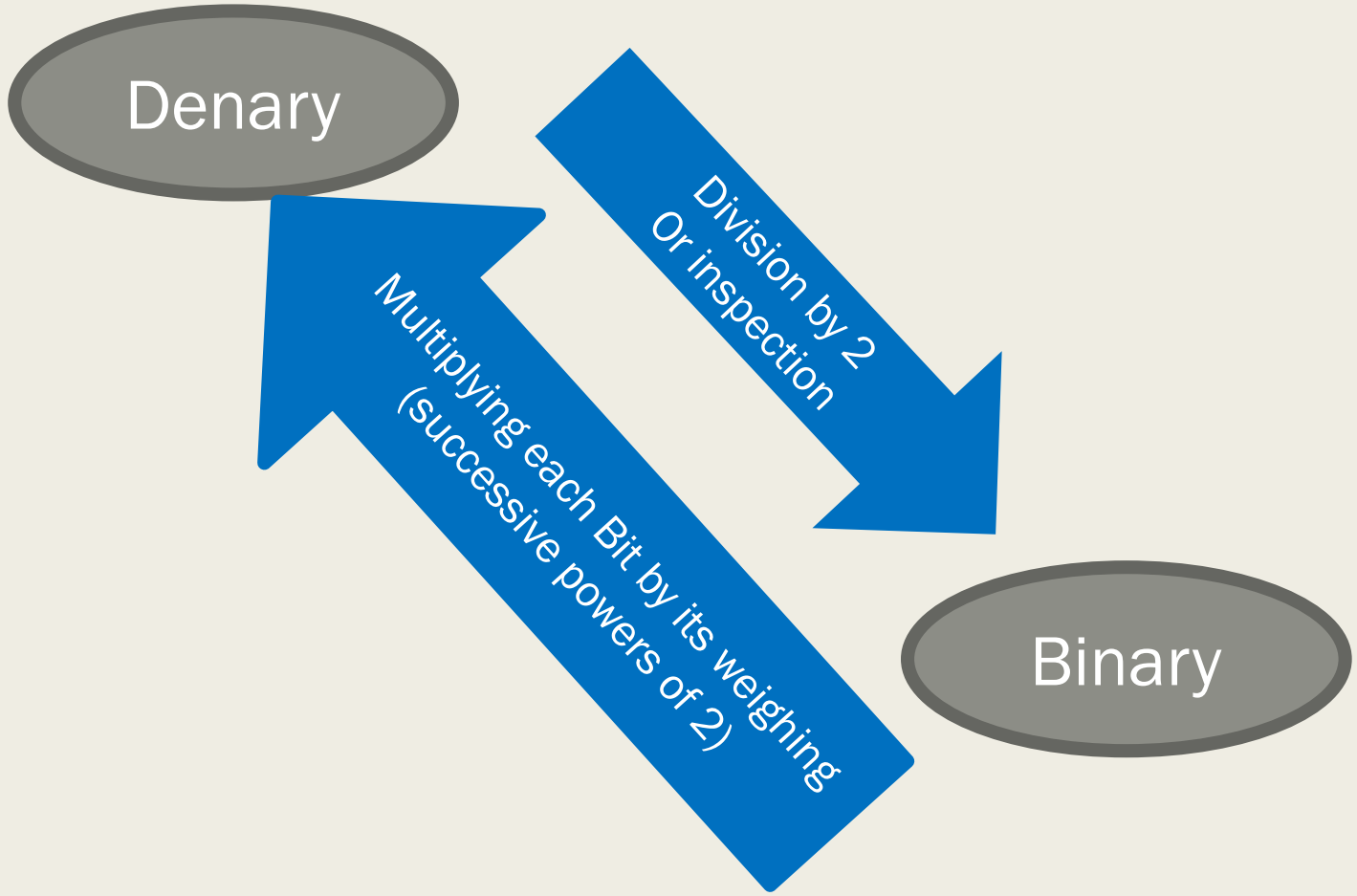
- 1 Hexadecimal digit
  - *Values: 0 → 15 (0, 1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 ,9 ,A ,B ,C ,D ,E ,F)*
    - $0 \rightarrow (16^1 - 1)$
  - 16^1 *values*
  - *Nibble*
- 2 Hexadecimal digit
  - *Values: 0 → 255*
    - $0 \rightarrow (16^2 - 1)$
  - 16^2 *values*
- *n* Hexadecimal digit
  - *Values: 0 →* $(16^n - 1)$
  - 16^n *values*

Why binary?

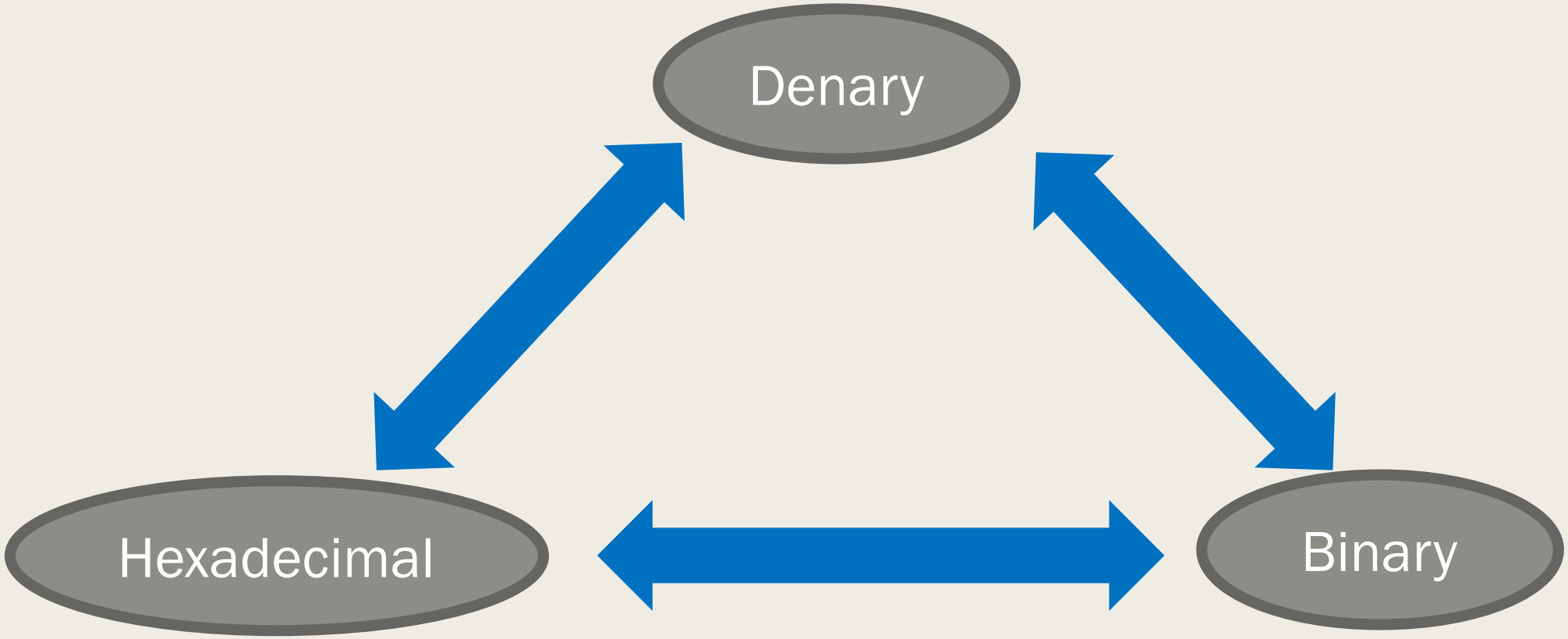
- Computers use Boolean Logic
- Boolean logic gates are based on a two-state system
- Transistors hold one of two states
- Many millions of transistors on each chip
- Can be read and set quickly

- But not very human readable
- Large numbers have lots of bits of data

Number System Triangle:



Number System Triangle:



Why Hexadecimal ?

- More efficient than Decimal for large numbers
- Quick conversion to / from Binary
  - *Nibble*
  - *16 is* 2^4
- Byte
  - *8 Bits*
  - *2 Nibbles*
  - *2 Hex digits*

2				A				B				3			
0	0	1	0	1	0	1	0	1	0	1	1	0	0	1	1

Binary				Hexadecimal
			0	0
			1	1
		1	0	2
		1	1	3
	1	0	0	4
	1	0	1	5
	1	1	0	6
	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

4096 table	
1	4096
2	8192
3	12288
4	16384
5	20480
6	24576
7	28672
8	32768
9	36864
10	40960
11	45056
12	49152
13	53248
14	57344
15	61440

256 table	
1	256
2	512
3	768
4	1024
5	1280
6	1536
7	1792
8	2048
9	2304
10	2560
11	2816
12	3072
13	3328
14	3584
15	3840

16 table	
1	16
2	32
3	48
4	64
5	80
6	96
7	112
8	128
9	144
10	160
11	176
12	192
13	208
14	224
15	240

In this video we looked at:

- Ranges of values
  - *Positive Integers*
    - Decimal
    - Binary
    - Hex
  - *Why use Binary?*
  - *Why use Hexadecimal?*