Digital Circuit Design

Li Bai

Module 4

- Number conversions
- BCD (Binary coded Decimal), ASCII (American Standard Code for Information Interchange), etc.
  - UTF (Unicode Transformation Format-7=ASCII)
  - UTF-16
- Binary addition
- https://www.edaplayground.com/x/bDf
- https://www.chipverify.com/verilog/verilog-display-tasks

- Number systems
    - e.g. base 7 days per week, base 24 hours, …
    - Decimal (base 10)
    - Binary (base 2)
    - Hexadecimal (base 16)
- Conversions between number systems

Numer system rules

- In a base n system, n basic symbols or n digits are used.
- The value of a number is determined by the symbols and their positions within the description.
- For integers, the starting number (symbol) is zero (0).
- This class will initially focus on integer representations and not fractions.

Number systems

- Decimal
  - base: 10
  - symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Binary
  - base: 2
  - symbols 0, 1
- Hexadecimal
  - base: 16
  - symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Number Systems' Notations

- Decimal
  - Nothing/Default: 5,487
  - Subscript: $5{,}487_{10}$
- Binary
  - Subscript: 1011_2
  - Prefixes (e.g., Verilog): 4'b1011
- Hexadecimal
  - Subscript: 73EA16
  - Prefixes: (C and others) 0x73EA
  - Prefix - Verilog 8'hf4 (8 refers to the number of bits, not digits)

Decimal Number System

Position	3	2	1	0
Multiplier	1000	100	10	1
	10^3	10^2	10^1	10^0
Example	5	4	8	7

5 x 1,000 =	5,000
4 x 100 =	400
8 x 10 =	80
7 x 1 =	7
Number =	5,487

Binary Number System

Position	3	2	1	0
Multiplier	8	4	2	1
	2^3	2^2	2^1	2^0
Example	1	0	1	1

1 x 8 =	8
0 x 4 =	0
1 x 2 =	2
1 x 1 =	1
Number =	11

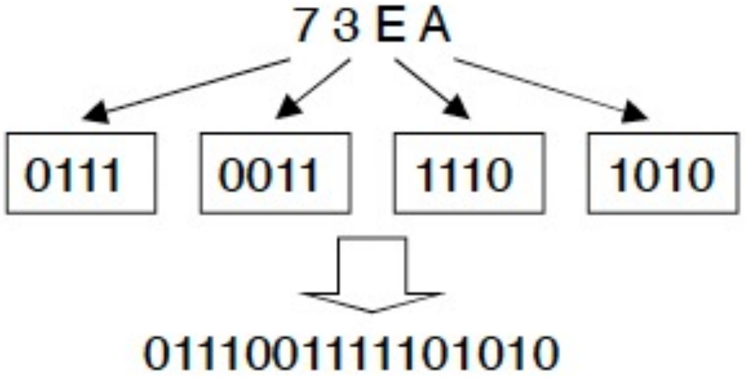
Hexadecimal Number System

Position	3	2	1	0
Multiplier	4,096	256	16	1
	16^3	16^2	16^1	16^0
Example	7	3	E	A

7 x 4,096 =	28,672
3 x 256 =	768
14(E) x 16 =	224
10(A) x 1 =	10
Number =	29,674

Conversion: Hex to Binary

- Recall the binary representation (4 bits) of the hexadecimal digits and expand one digit at a time
- Example: 0x73EA



7 3 E A

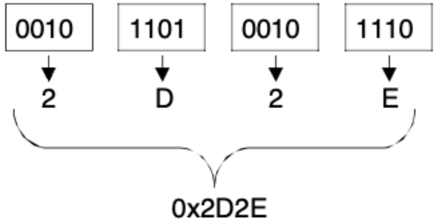
0111 0011 1110 1010

0111001111101010

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

Conversion: Binary to Hex

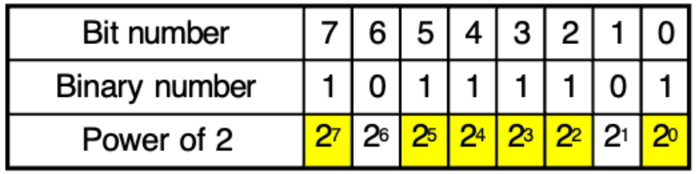
- Starting with the lsb, slice the binary number into segments of 4 bits
- Recall the hexadecimal representation of the binary segments
- Example: 0010110100101110



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

Conversion: Binary to Decimal

- Sum over the powers of 2n where n is the bit number, starting by counting the least significant bit as number 0.

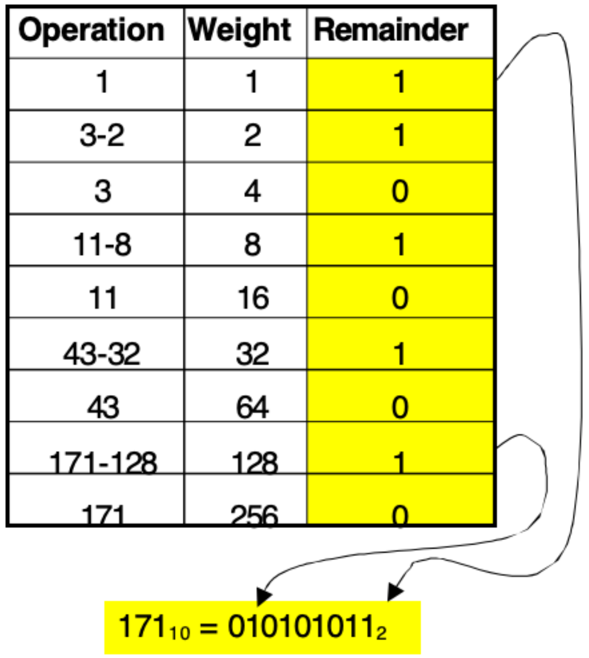


Bit number	7	6	5	4	3	2	1	0
Binary number	1	0	1	1	1	1	0	1
Power of 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

128 + 32 + 16 + 8 + 4 + 1 = 189

Conversion: Decimal to Binary

- Get binary weights for each bit, compare to the largest and subtract.
- Example: 171_{10}

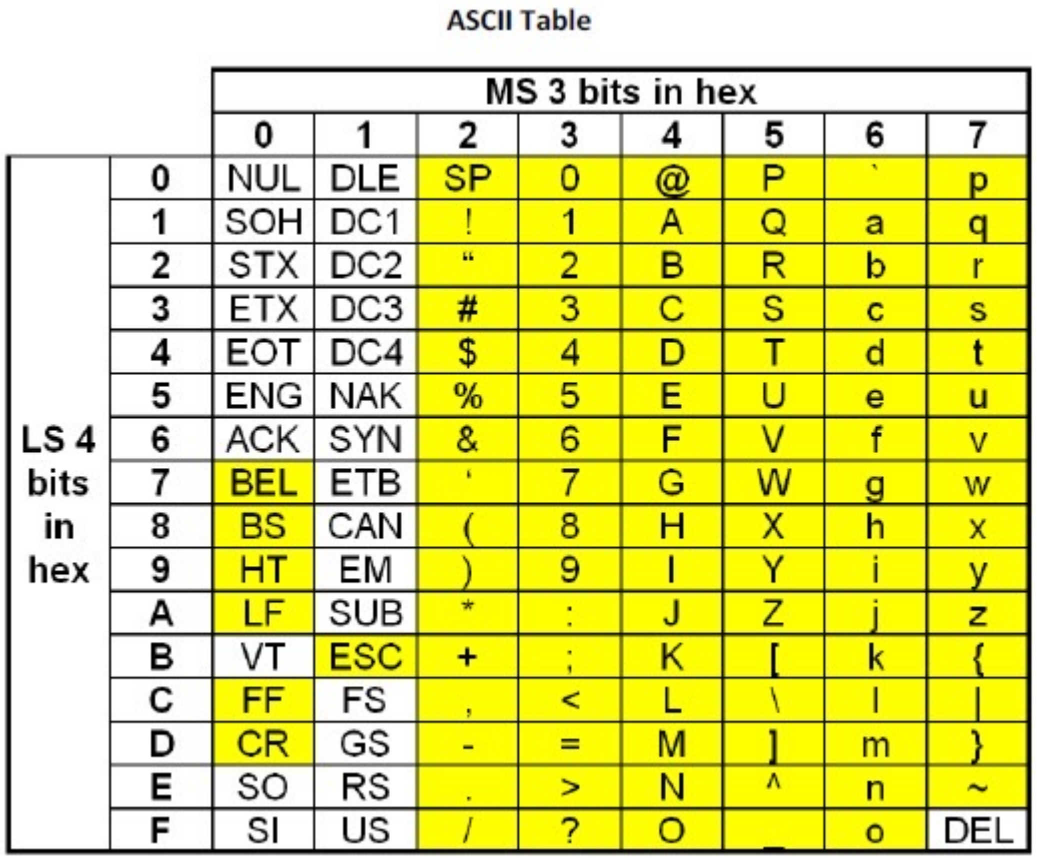


Operation	Weight	Remainder
1	1	1
3-2	2	1
3	4	0
11-8	8	1
11	16	0
43-32	32	1
43	64	0
171-128	128	1
171	256	0

$171_{10} = 010101011_2$

https://www.edaplayground.com/x/uuR5

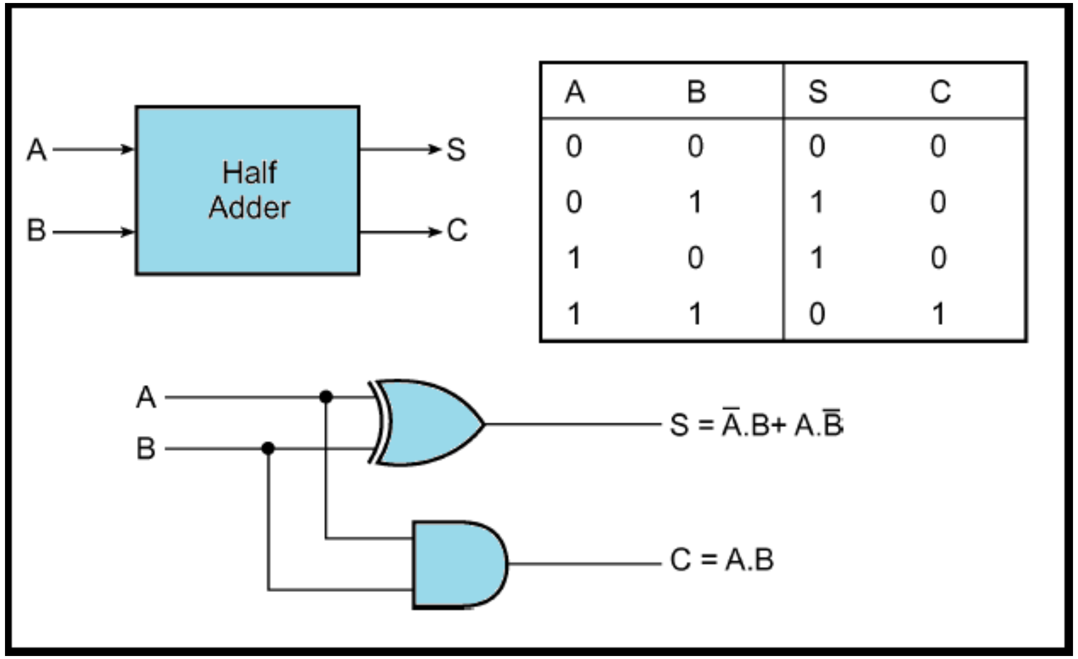
ASCII table



ASCII Table

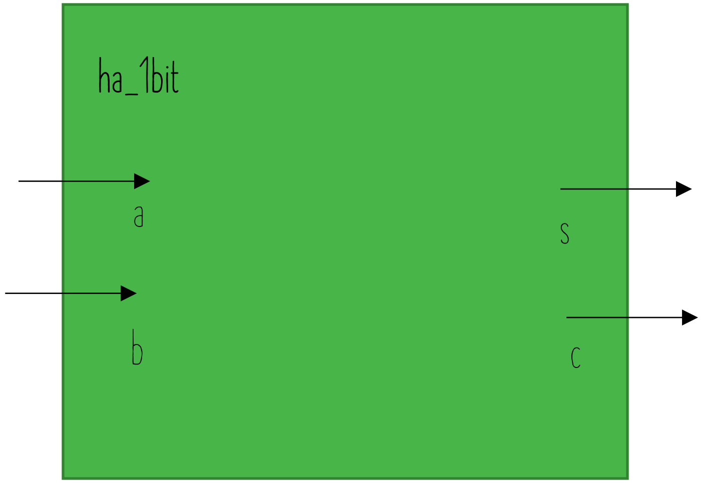
		MS 3 bits in hex							
		0	1	2	3	4	5	6	7
LS 4 bits in hex	0	NUL	DLE	SP	0	@	P	`	p
	1	SOH	DC1	!	1	A	Q	a	q
	2	STX	DC2	“	2	B	R	b	r
	3	ETX	DC3	#	3	C	S	c	s
	4	EOT	DC4	$	4	D	T	d	t
	5	ENG	NAK	%	5	E	U	e	u
	6	ACK	SYN	&	6	F	V	f	v
	7	BEL	ETB	‘	7	G	W	g	w
	8	BS	CAN	(	8	H	X	h	x
	9	HT	EM	)	9	I	Y	i	y
	A	LF	SUB	*	:	J	Z	j	z
	B	VT	ESC	+	;	K	[	k	{
	C	FF	FS	,	<	L	\	l	|
	D	CR	GS	-	=	M	]	m	}
	E	SO	RS	.	>	N	^	n	~
	F	SI	US	/	?	O	_	o	DEL

Half adder



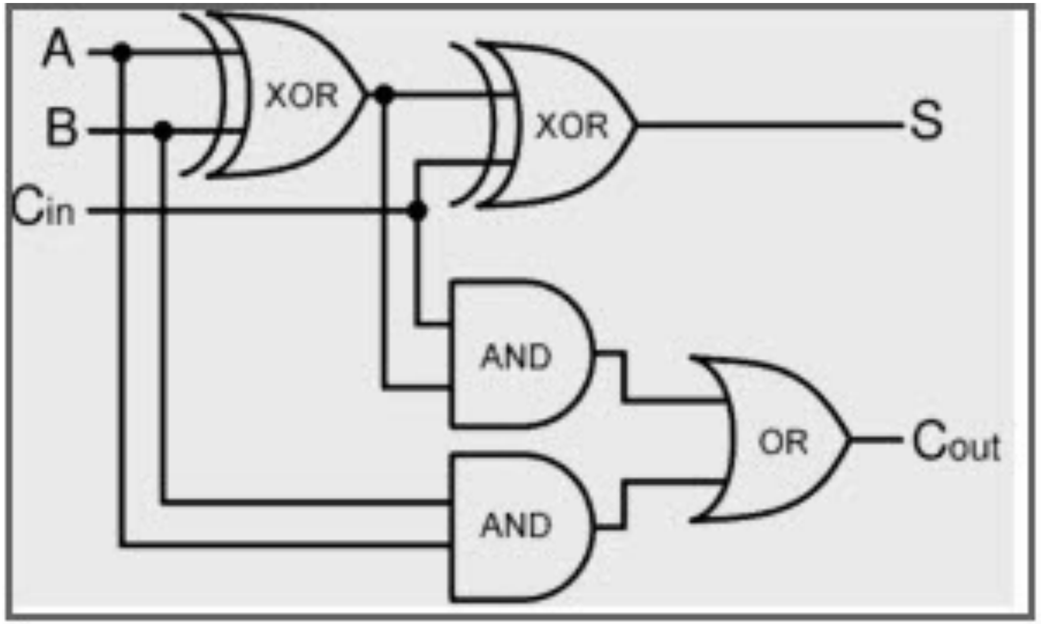
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

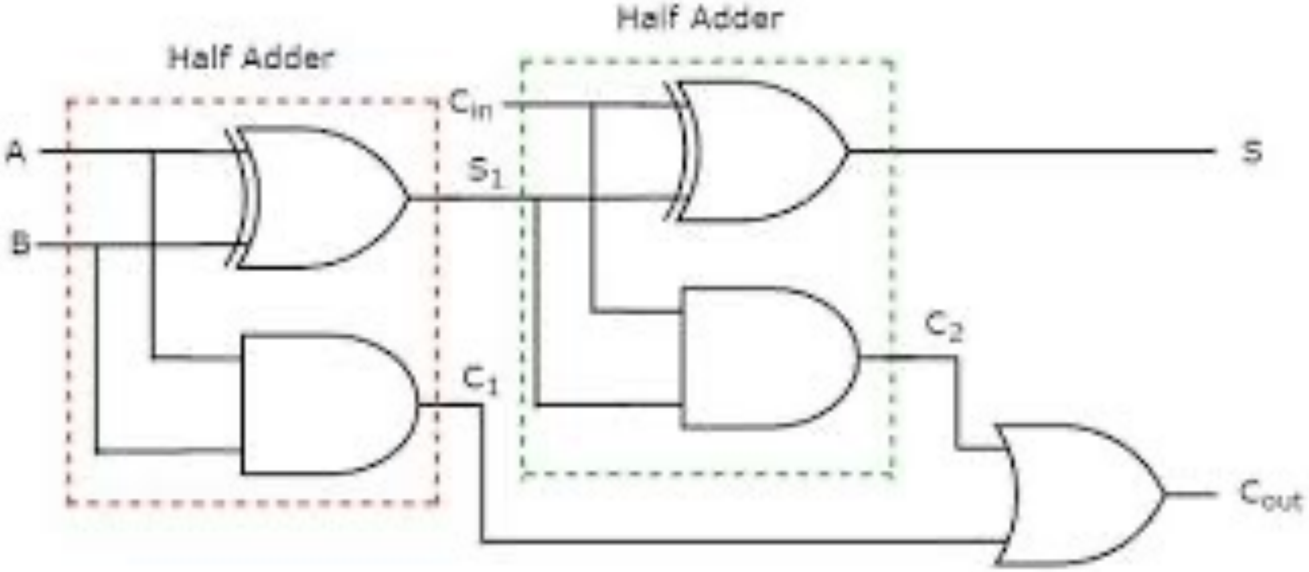
https://www.edaplayground.com/x/KSce



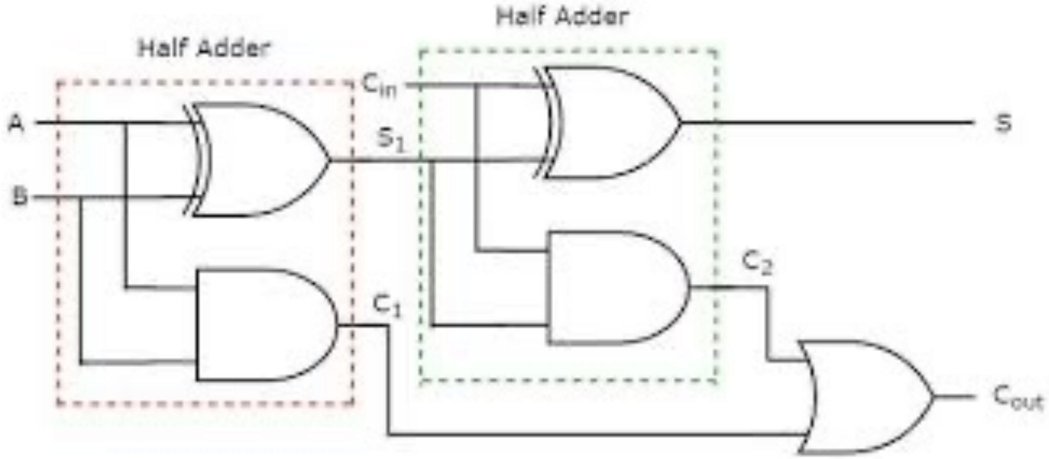
Binary addition

Fulladder

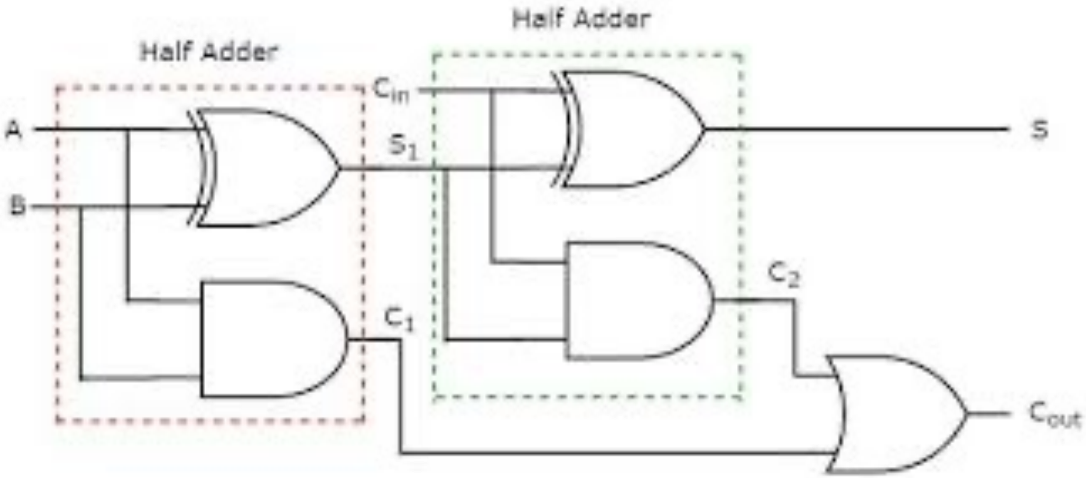

A
B
Cin
XOR
XOR
S
AND
AND
OR
Cout


Half Adder
Half Adder
A
B
C_{in}
S_1
S
C_1
C_2
C_{out}

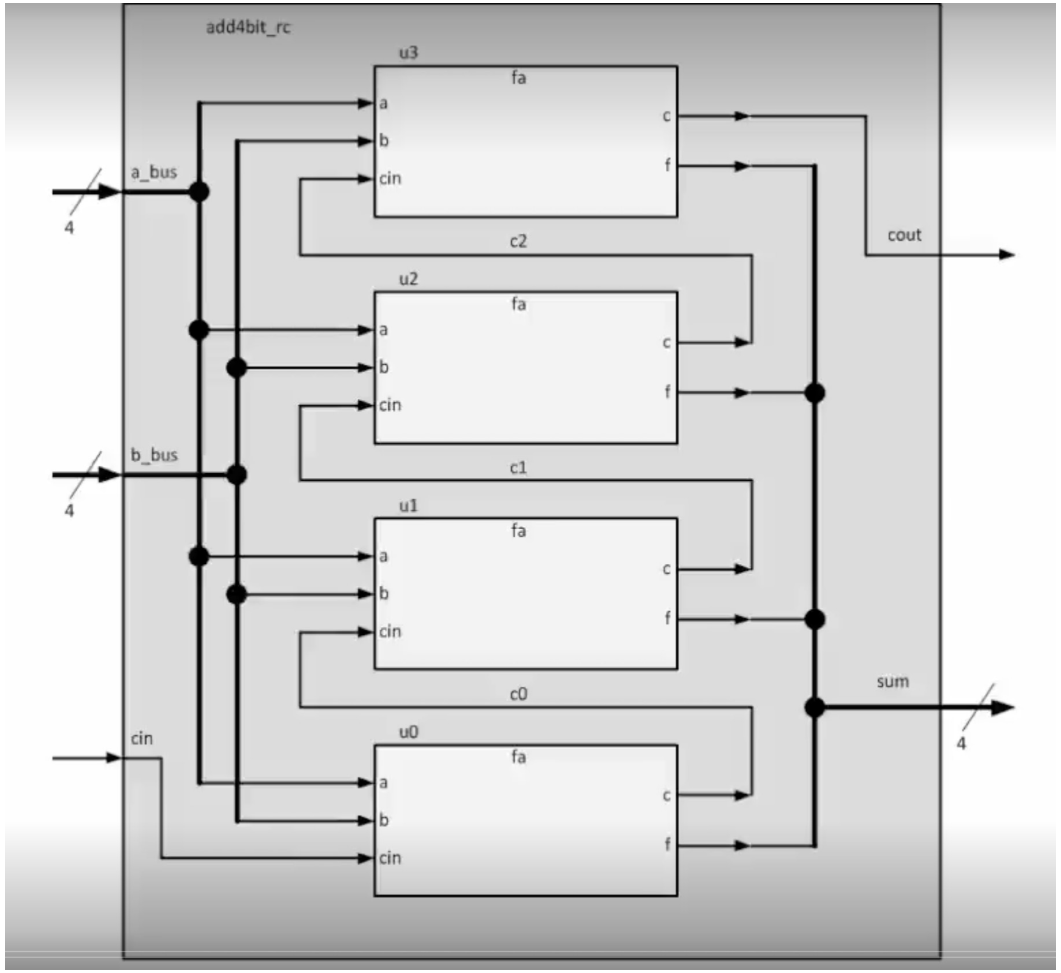
Fulladder - increment



Let B=0
Cin = 0



verilog excercise



https://www.edaplayground.com/x/d46m