SRA

Project : RISC-V based CPU : Apexlore

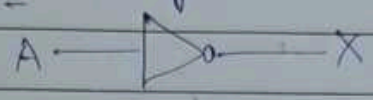
Learnings :-

(1) Digital logic :

a) Combinational logic

Gates :-

→ NOT

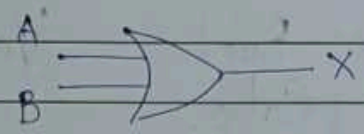


A	X
0	1
1	0

→ AND

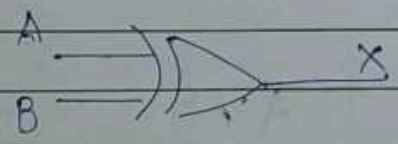
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

→ OR



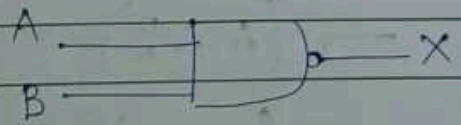
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

→ XOR



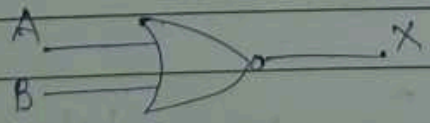
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

→ NAND



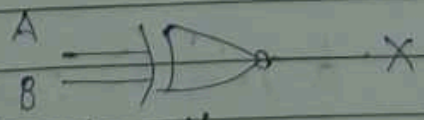
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

→ NOR



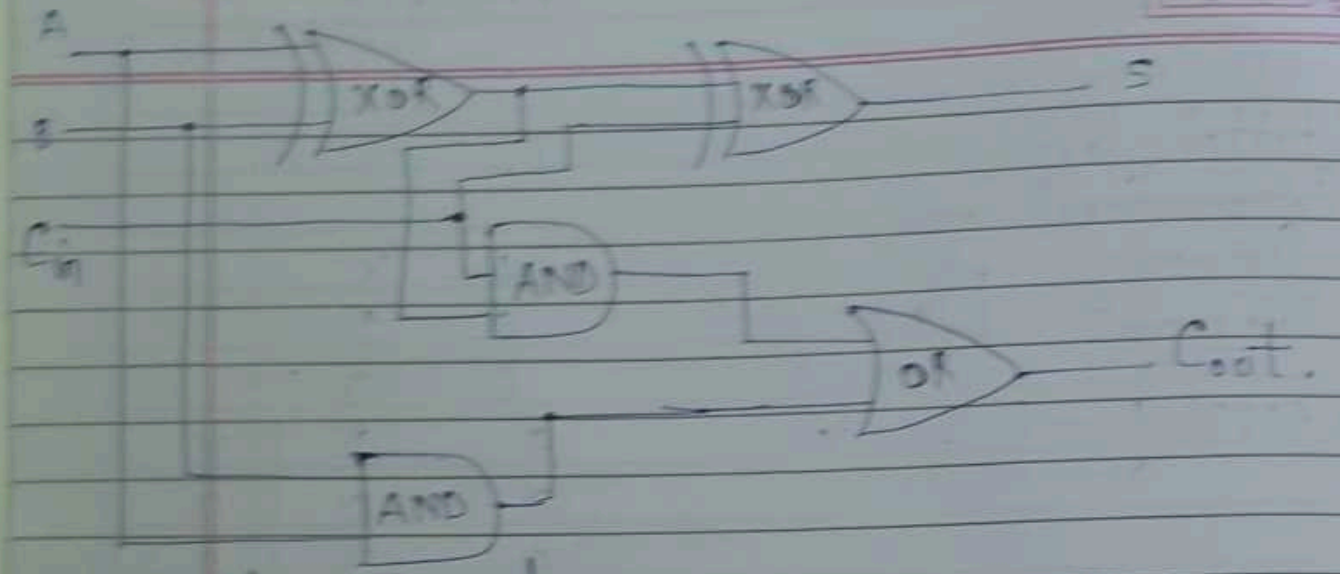
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

→ XNOR



A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

Full-adder



II Verilog Operators

Operation	Bool Arith.	Bool Calc.	Verilog code
NOT	$\overline{A}$	$\neg A$	~A or !A
AND	$A \cdot B$	$A \wedge B$	A&B or &&
OR	$A + B$	$A \vee B$	A|B or ||
XOR	$A \oplus B$	$A \oplus B$	A ^ B
NAND	$\overline{A \cdot B}$	$\neg(A \wedge B)$	!(A&B)
NOR	$\overline{A + B}$	$\neg(A \vee B)$	!(A|B)
XNOR	$\overline{A \oplus B}$	$\neg(A \oplus B)$	!(A^B)

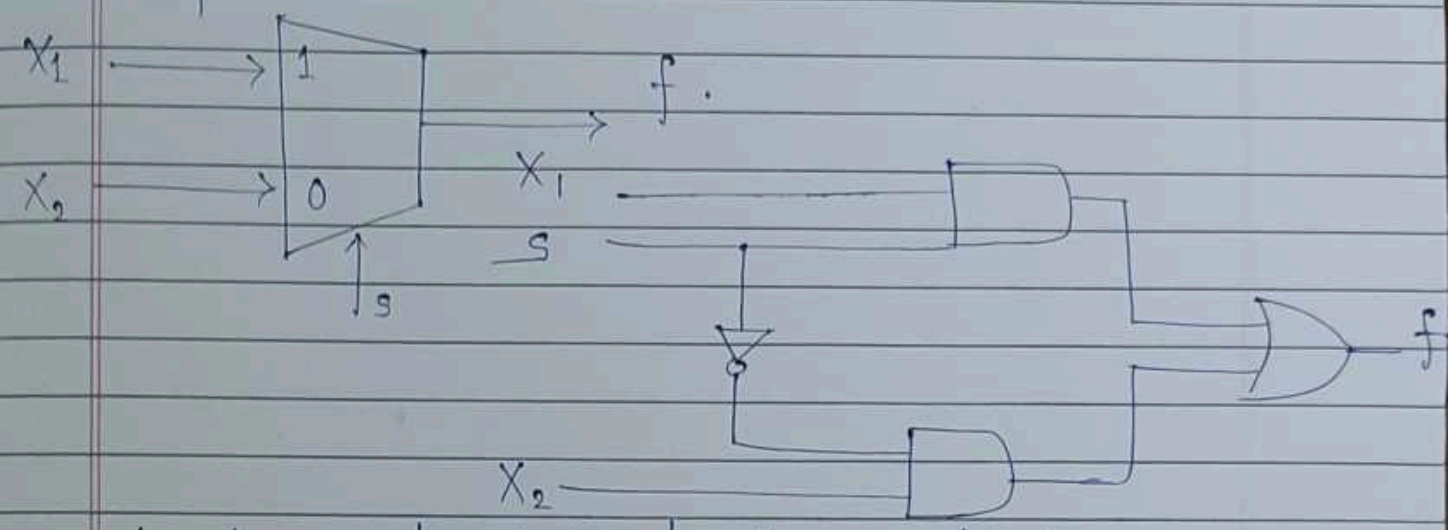
b) Common data-types :-
booleans and bit vectors.

vector declaration : $ vecl [7 : 0] = ...;

operations : +, -, *, /, %

==, !=, >, >=, <, <=

c) Multiplexers




```
$out = $sel ? $in1 : $in0;
```

→ $out is : if $sel then $in1
otherwise $in0

```
$out[7:0] = $sel[3] ? $in3 : $sel[2] ?
            $in2 : $sel[1] ? $in1 :
            $in0;
```

8-bit wide multiplexer. (top-to-bottom)

d) Literals

```
$foo[7:0] = 6;
          = 8'd6;
          = 8'b110;
          = 8'h6
```

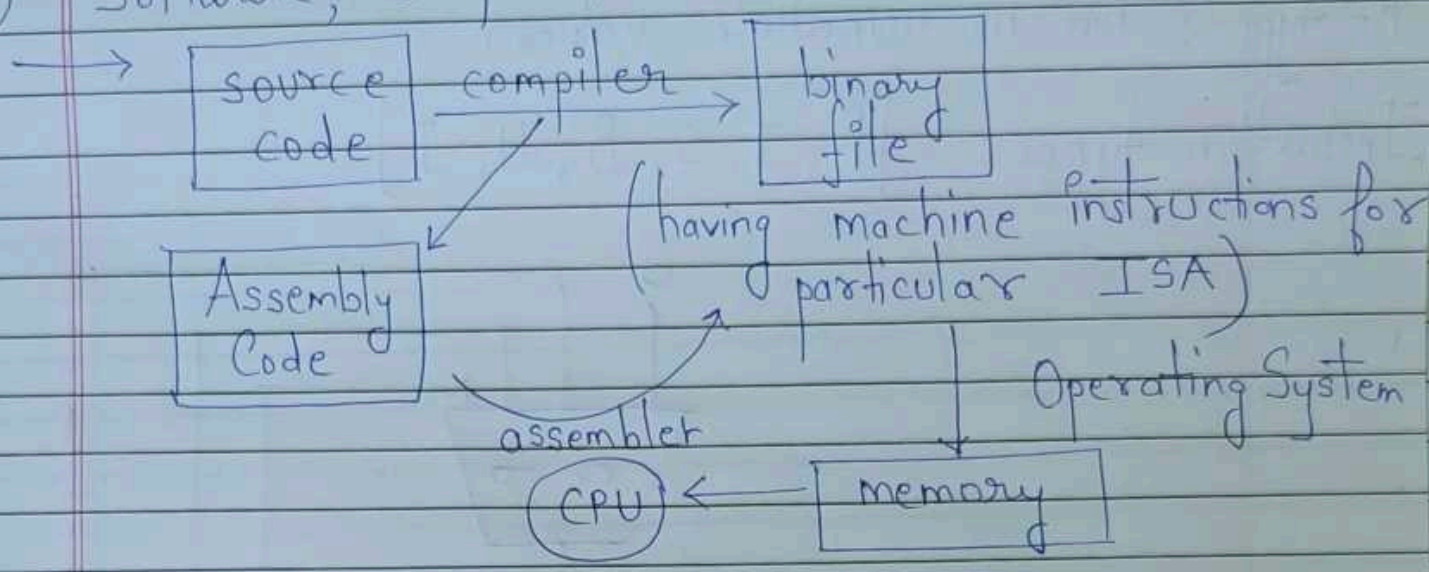
6 coerced to 8 bits

e) Concatenation

```
$word[15:0] = ($upper_byte, $lower_byte);
```

PAGE No.
DATE / /

① Software, Compiler, CPU



② RISC : Reduced Instruction Set Computing

- load-store architecture
- has register file of storing upto 32 values (31)
- instructions read from & write back to file.
→ btw memory and register.

fields

a) opcode
general classification of instruction & determine which of remaining fields are needed, how r they laid out, encoded in remaining instruction bits.

b) function field (funct3/ funct7)
→ specify exact f^n performed by instruction.

c) rs1/ rs2
indices (0-31) indentifying register in register files having operand values on which operation carried.

d) rd
index (0-31) of register into which result written

e) immediate
value btw bits. Value give offset for indexing

into memory or,
value upon which to operate (in place register value indexed by rs2)

R-type : has no immediate values.

Instruction types : R, I, S, B, U, J