Digital Logic: Boolean Algebra and Gates (contd..)

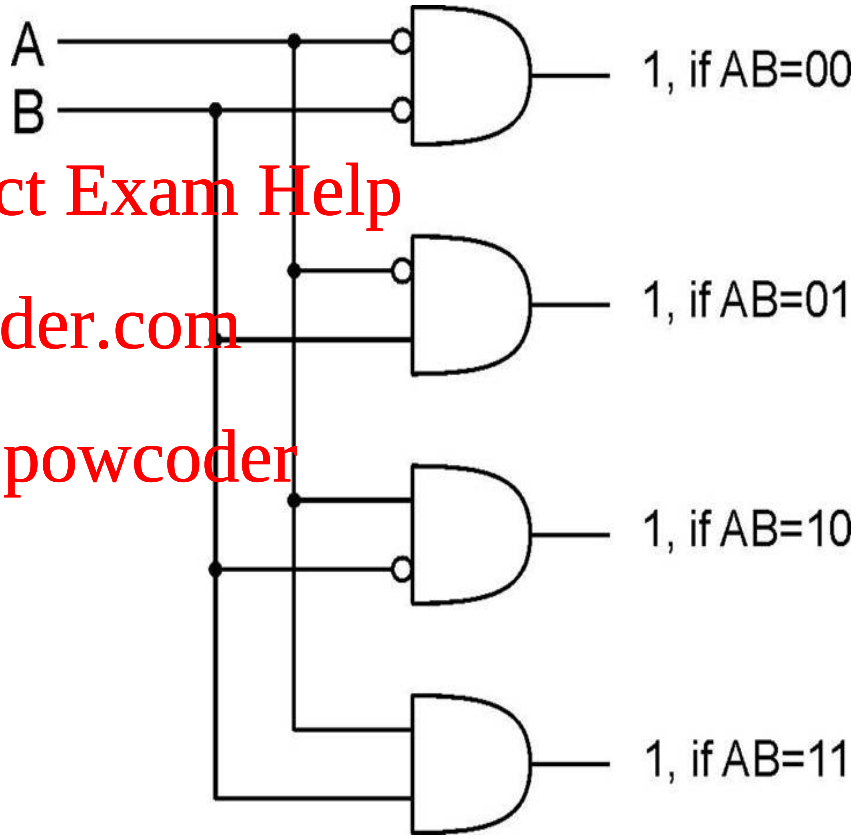
n-to-2^n Decoder

- *n* inputs, 2^n outputs
  - exactly one output is 1 for each possible input pattern
- Uses:
  - Convert memory or register address to a control line
  - Convert an opcode to one of n control lines
  - We will get to this in the MIPS material

Two-to-Four Decoder

TRUTH TABLE:

A	B	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



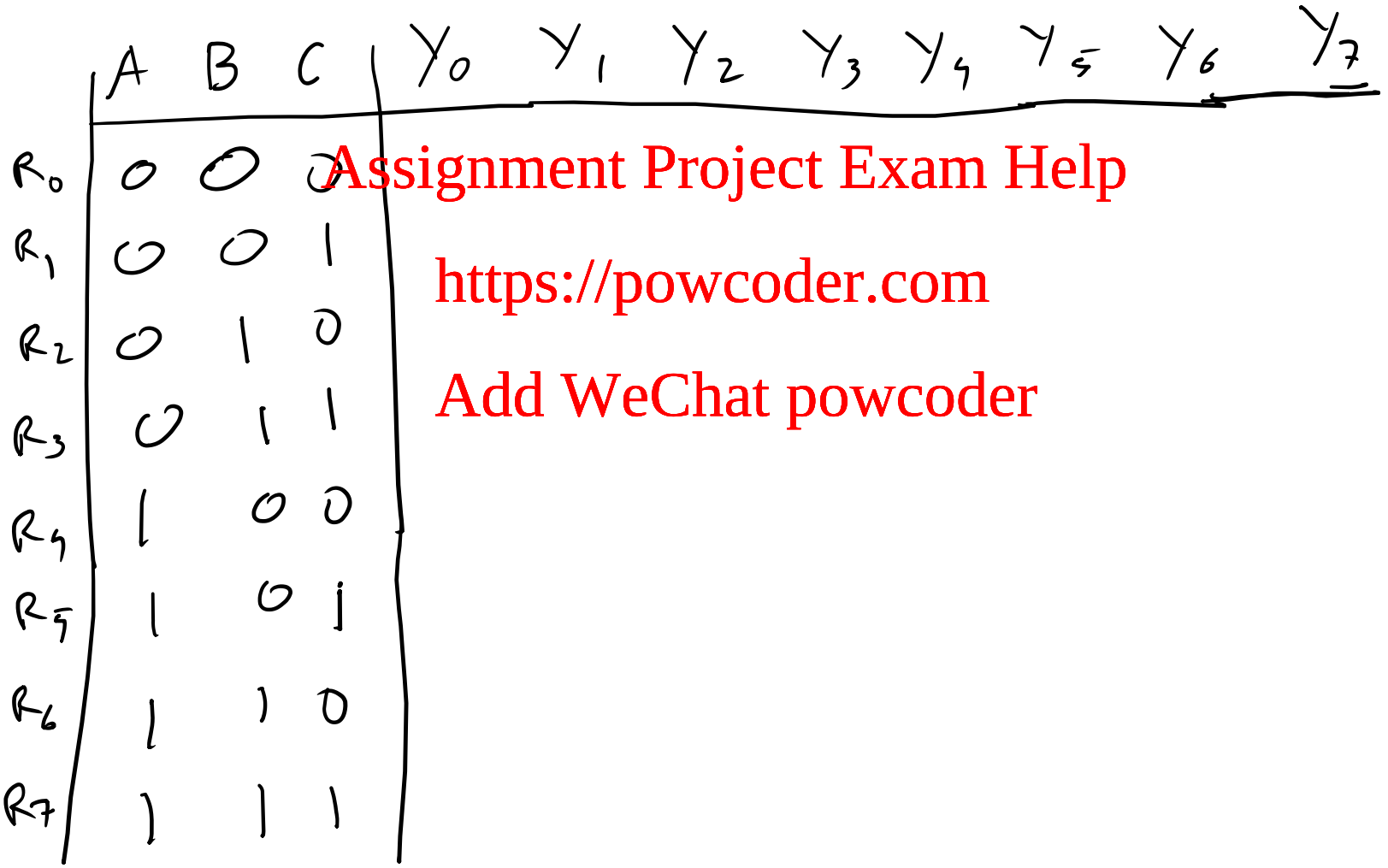
Three-to-Eight Decoder

$[\ 2^3 = 8\ ]$

	A	B	C	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
R_0	0	0	0								
R_1	0	0	1								
R_2	0	1	0								
R_3	0	1	1								
R_4	1	0	0								
R_5	1	0	1								
R_6	1	1	0								
R_7	1	1	1								

How to decode for 6 output lines?

- How many input lines do we need?



Time for some...

- We currently use decimal system in daily life (deci=10 digits,0-9)
- We know..

1+0=1

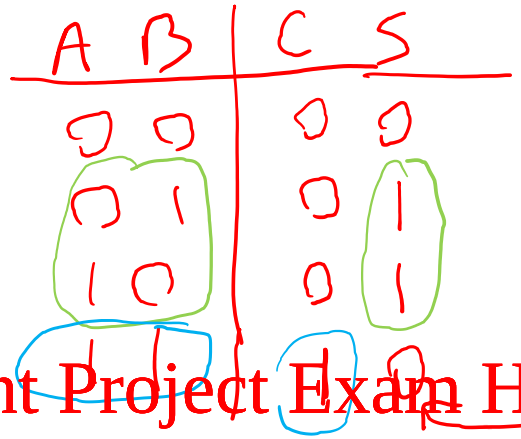
1+1=2;1+2=3;1+3=4…

1+8=9;

- What is 1+9=??

Binary Addition and Half-Adder

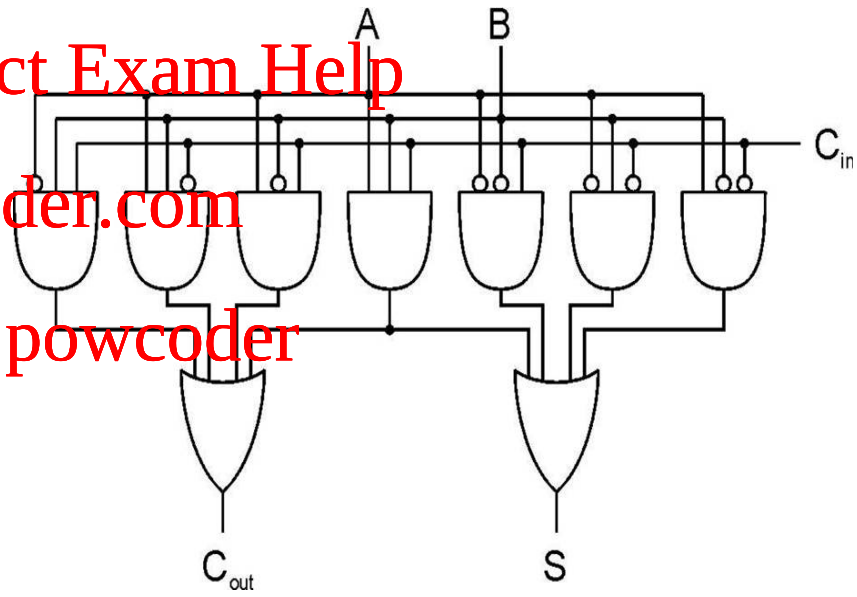
- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 10



- A half-adder can add 2 bits and produces a sum and carry signal
  - Sum = A xor B
  - Carry = AB

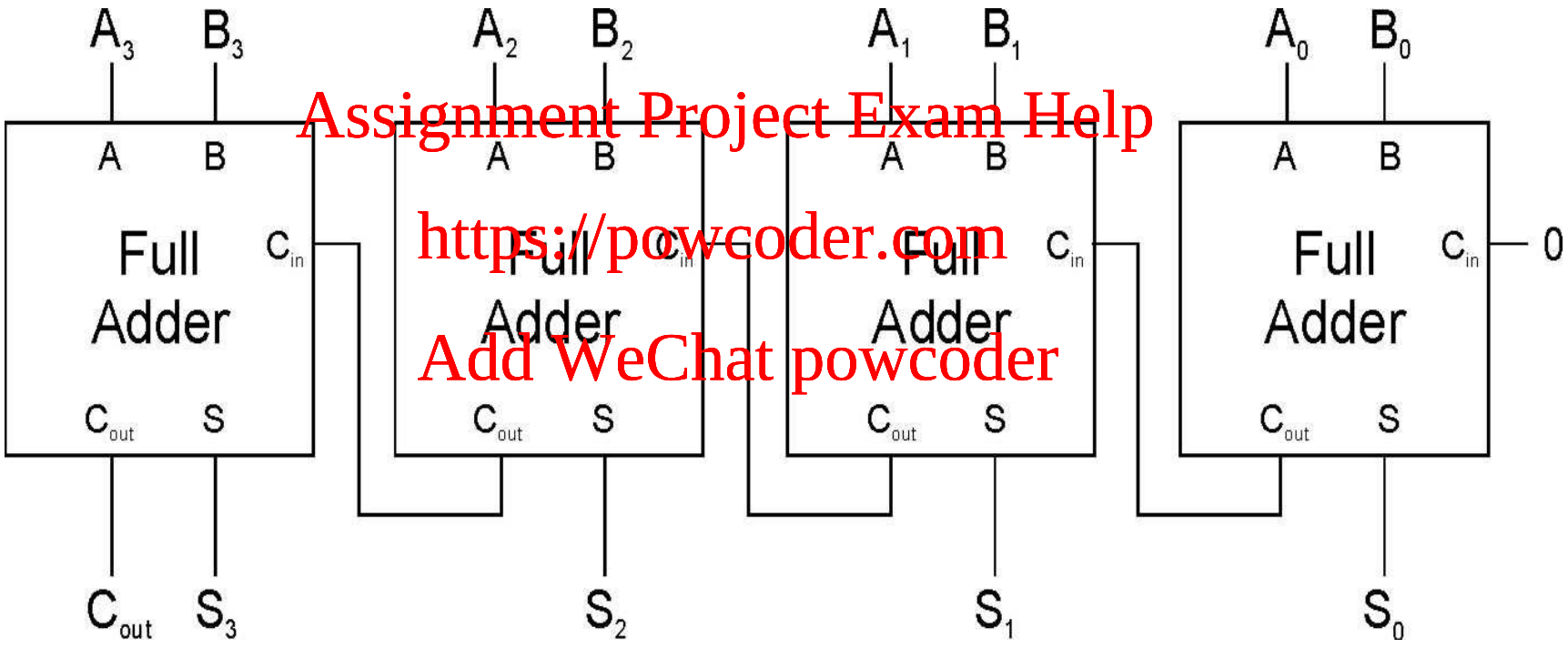
One-Bit Full Adder

A	B	C_{in}	C_{out}	S
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



Four-Bit Full Adder

Ripple-carry adder



Masking

- Want to look only at certain bits of a binary word
- Use a mask to remove the uninteresting bits
- Example:
  - Two values: 10011101 and 01001001
  - If we want to see bit 3 from right, we AND it with 00000100 to get
    - 00000100 and 00000000, respectively.

Logic Minimization (CSE 100!)

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Start with SOP:

A’BC’ + A’BC + AB’C’ + ABC’

Factor A’ and A out:

A’(BC’ + BC) + A(B’C’ + BC’)

Factor B and C’ out:

A’(B(C’ + C)) + A(C’(B’ + B))

Use identity that C’+C=1 and B’+B=1:

A’B + AC’

Building functions from logic gates

- Combinational Logic Circuit
  - Output depends only on the current inputs
  - Stateless (memoryless)
- Sequential Logic Circuit
  - Output depends on the sequence of inputs (past and present)
  - Stores information (state) from past inputs

Sequential Circuits and Memory