

Week 1 - Tuesday: Electronics - Digital Signals & Simple Circuits (Key)

Name: _____ Date: _____

Digital circuits use only TWO levels: LOW = 0 = about 0 V HIGH = 1 = 3.3 V or 5 V

Part A - Digital signals (SIM-001)

1) Fill in the table: one switch wired to one light.

Switch	Signal (0 or 1)	Light (on/off)
OFF	0	off
ON	1	on

2) What does 0 (LOW) mean in a digital circuit? About how many volts?

LOW = 0 = ground, about 0 volts. Off!

3) What does 1 (HIGH) mean? Name a HIGH voltage.

HIGH = 1 = the supply voltage, like 3.3 V or 5 V. On!

4) What does the word 'binary' mean?

Using only two states: 0 and 1.

Part A (continued)

5) Why are two states (0 and 1) so reliable? Give one reason.

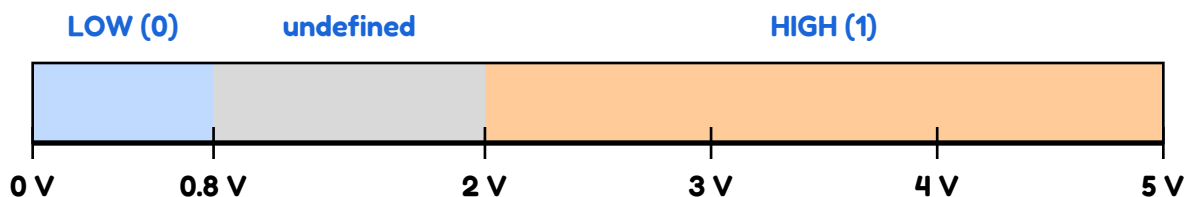
They are easy to tell apart, they resist noise, and transistors work like on/off switches.

6) Remember Monday! On the ESP32, `digitalWrite(2, HIGH)` puts about how many volts on the pin?

About 3.3 V - that is a 1, so the LED turns on. LOW = 0 V = off.

Part B - Logic levels and thresholds (SIM-002)

7) Color or label the zones: LOW (0 to 0.8 V), UNDEFINED (0.8 to 2.0 V), HIGH (2.0 to 5.0 V).



Part B (continued) - HIGH, LOW, or undefined?

Rules: 2.0 V or more = HIGH. 0.8 V or less = LOW. In between = undefined.

8) 0.2 V ->

LOW (0)

9) 4.5 V ->

HIGH (1)

10) 1.5 V ->

undefined

11) 3.3 V ->

HIGH (1)

12) 0.8 V ->

LOW (0) (on the edge)

13) 1 V ->

undefined

14) What does a BUF (buffer) do?

It makes a weak signal strong and clean again, gives extra current to drive a light, and adds a tiny delay.

15) Noise margin: a HIGH output is at least 2.4 V and an input needs 2.0 V. $2.4 - 2.0 = ?$ volts of safety.

0.4 V - that much noise can sneak in and it still reads as 1.

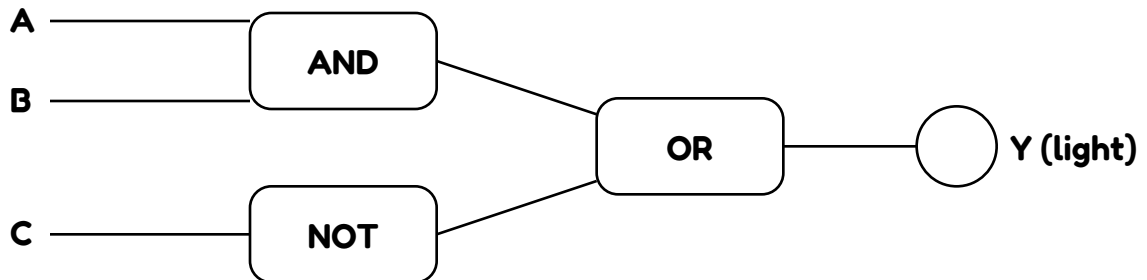
16) Is a BIGGER noise margin better or worse? Why?

Better - more protection from noise, so a 1 stays a 1 and a 0 stays a 0.

Part C - Building simple circuits (SIM-003)

AND = BOTH inputs must be 1. OR = at least ONE input is 1. NOT = flips it (0 -> 1, 1 -> 0).

17) Look at the circuit. Write the rule for light Y using AND, OR and NOT.



$$Y = (A \text{ AND } B) \text{ OR } (\text{NOT } C)$$

18) NOT gate table:

IN	OUT
0	1
1	0

19) AND gate table:

A	B	OUT
0	0	0
0	1	0
1	0	0
1	1	1

20) OR gate table:

A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	1

Part C (continued)

21) Full truth table: $Y = (A \text{ AND } B) \text{ OR } (\text{NOT } C)$. Fill in every Y!

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

22) Try it checks: what does light Y show? Write 1 (on) or 0 (off).

a) $A=1, B=1, C = \text{anything}$ $\rightarrow Y = \underline{\hspace{2cm}}$ 1 (on)

b) $A=0, B=0, C=0$ $\rightarrow Y = \underline{\hspace{2cm}}$ 1 (on)

c) $A=0, B=0, C=1$ $\rightarrow Y = \underline{\hspace{2cm}}$ 0 (off)

Part D - Try it on DigiSim!

23) Tick each box when you finish the step.

- ☒ 1. Open DigiSim SIM-001 Digital Signals
- ☒ 2. Press Play
- ☒ 3. Flip the switch ON - watch the light
- ☒ 4. Flip the switch OFF - watch the light
- ☒ 5. Open DigiSim SIM-002 Logic Levels
- ☒ 6. Press Play
- ☒ 7. Flip the switch - the signal goes through BUF
- ☒ 8. Watch the light turn on and off
- ☒ 9. Open DigiSim SIM-003 Building Simple Circuits
- ☒ 10. Press Play
- ☒ 11. Try A=1, B=1 (any C) - light Y is ON
- ☒ 12. Try A=0, B=0, C=0 - light Y is ON
- ☒ 13. Try A=0, B=0, C=1 - light Y is OFF

24) Explain: Why do computers use only 0 and 1?

Two states are easy to tell apart, even with noise. Transistors inside chips work like tiny on/off switches, so on = 1 and off = 0. Lots of 0s and 1s together can stand for numbers, letters and pictures.