

Lab 8: Digital Modulation

SUMMER CHALLENGE COURSE

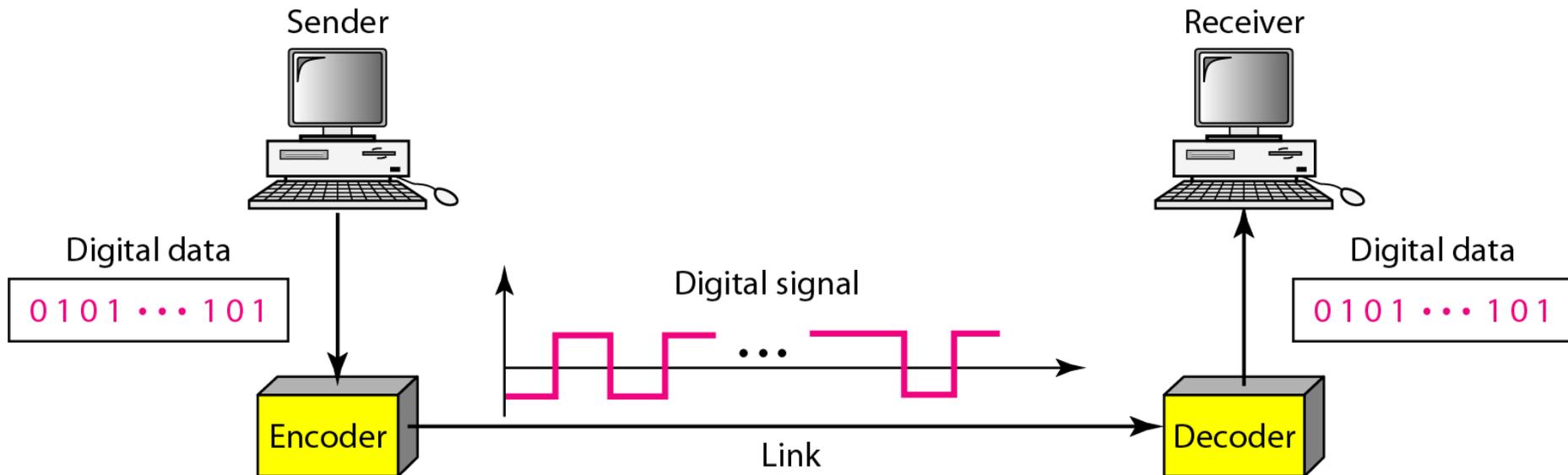
SMART LIGHTING

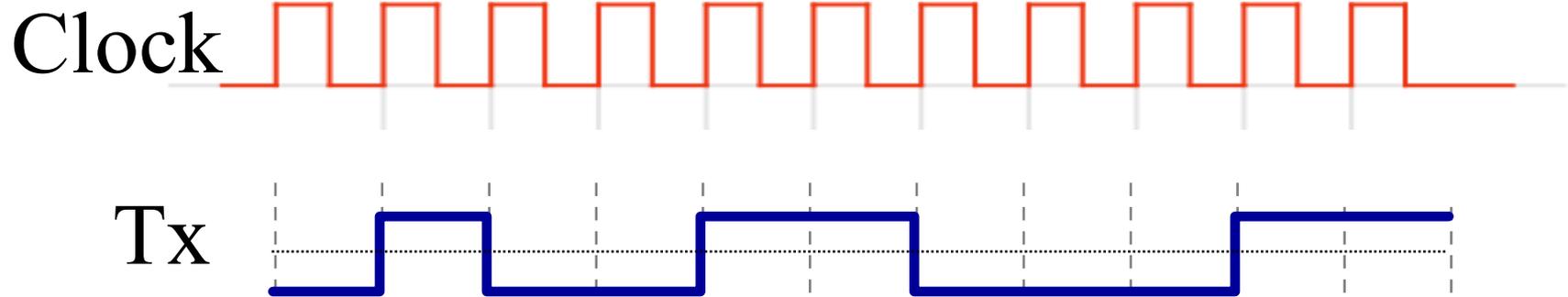
07/30/2014

Ayse Coskun

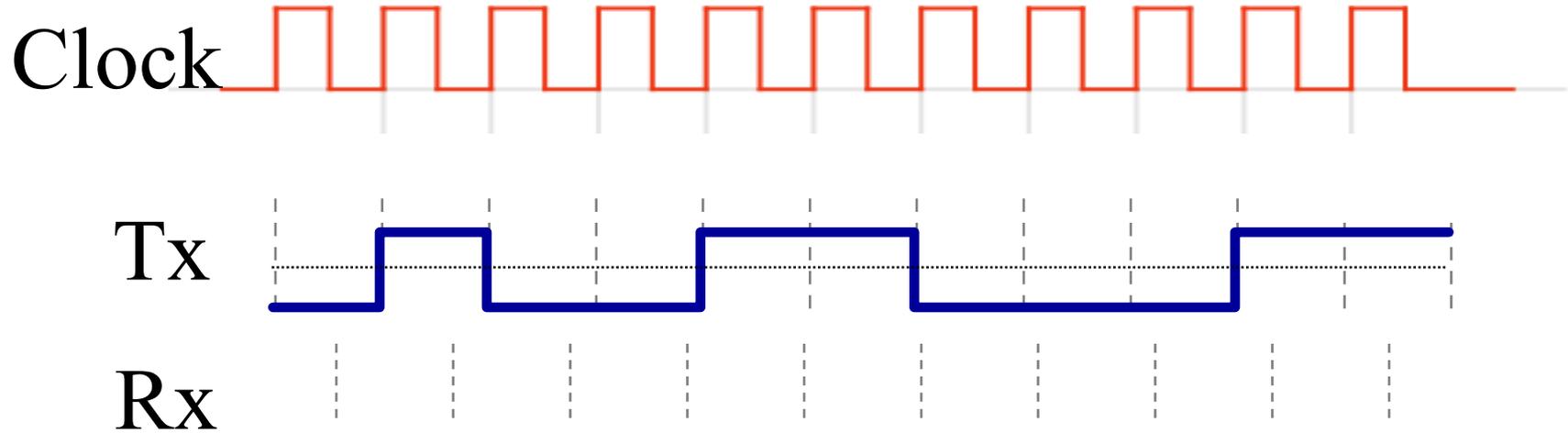
acoskun@bu.edu

- Converting a string of 1's and 0's (digital data) into a sequence of signals that denote the 1's and 0's.
- For example a high voltage level (+V) could represent a "1" and a low voltage level (0 or -V) could represent a "0".

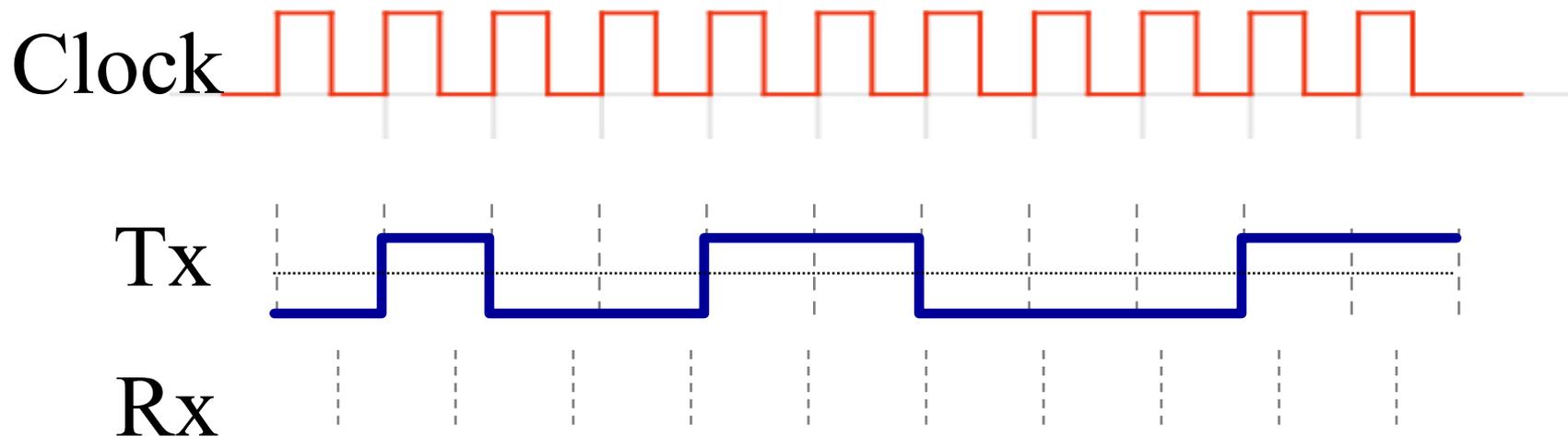




= power on (signal)
0 = power off (no signal)



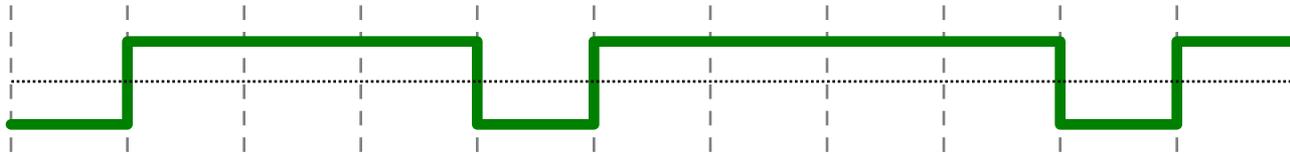
= power on (signal)
0 = power off (no signal)



1 = power on (signal)
0 = power off (no signal)

Problem(s)

- lack of “clock” recovery during long string of 0 or 1 bits
- “baseline wander” during long string of 0 or 1 bits

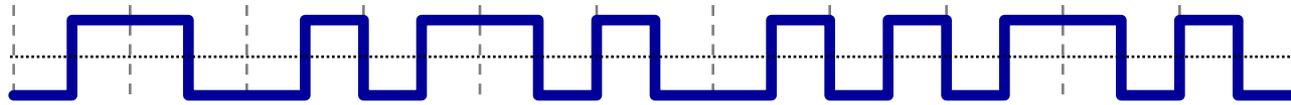


= change of signal level (on-off or off-on)
0 = no change of signal level

- NRZI is an example of **differential encoding**
- Fixes clocking problem for long string of bits

Problem(s)

- Lack of clock recovery during long string of bits



Always transition in middle of bit period:

0 = low-to-high transition

1 = high-to-low transition

- Good clock recovery
- How to implement this?

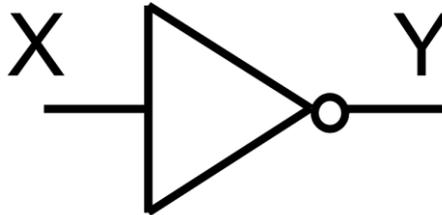
- **Boolean Functions (Logic Functions):** are function that return truth values; variables can only be 1 (true) or 0 (false).

NOT function: $Y = \sim X$

- **Boolean Functions (Logic Functions):** are function that return truth values; variables can only be 1 (true) or 0 (false).

NOT function: $Y = \sim X$

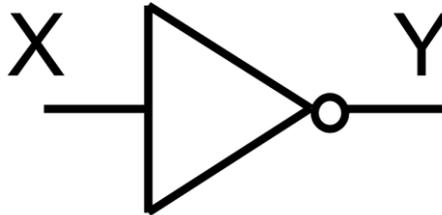
- **A Logic Gate:** is a physical device implementing a Boolean function.



- **Boolean Functions (Logic Functions):** are function that return truth values; variables can only be 1 (true) or 0 (false).

NOT function: $Y = \sim X$

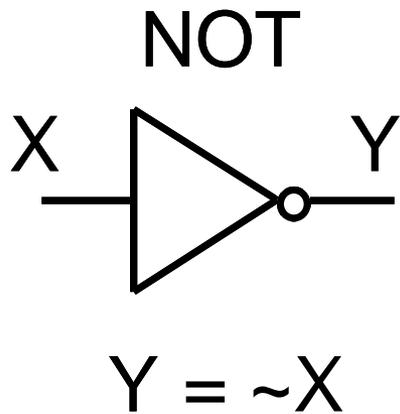
- **A Logic Gate:** is a physical device implementing a Boolean function.



- **A Truth Table:** shows how a logic circuit's output responds to inputs.

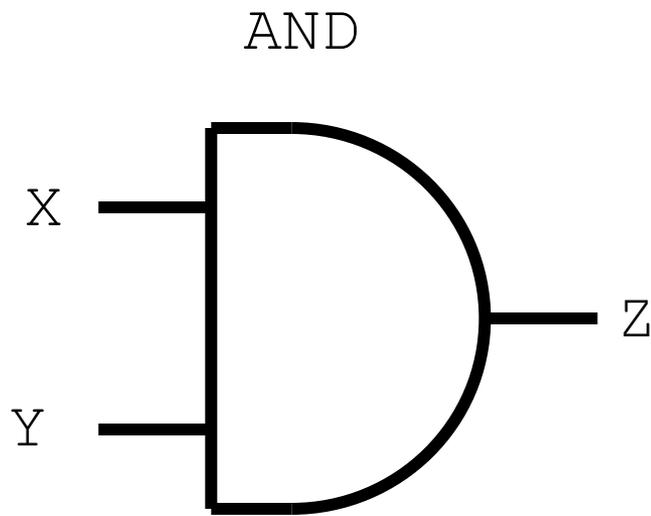
X	Y
0	1
1	0

NOT Gate -- Inverter



X	Y
0	1
1	0

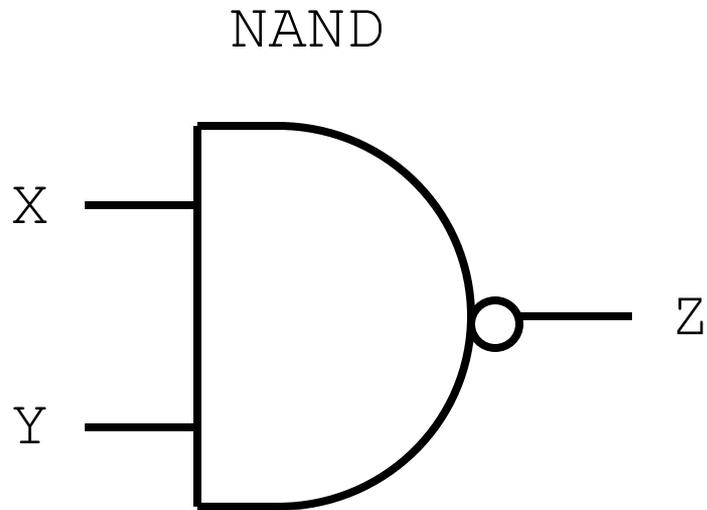
AND Gate



X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

$$Z = X \& Y$$

NAND Gate

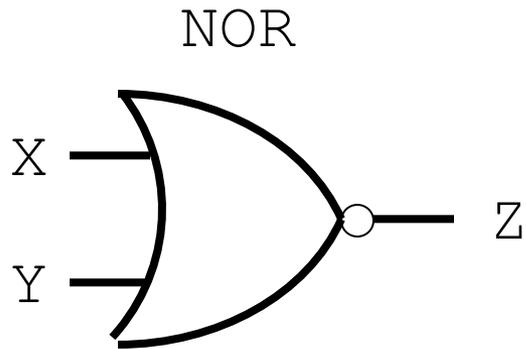


X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

$$Z = \sim (X \& Y)$$

nand(Z, X, Y)

NOR Gate

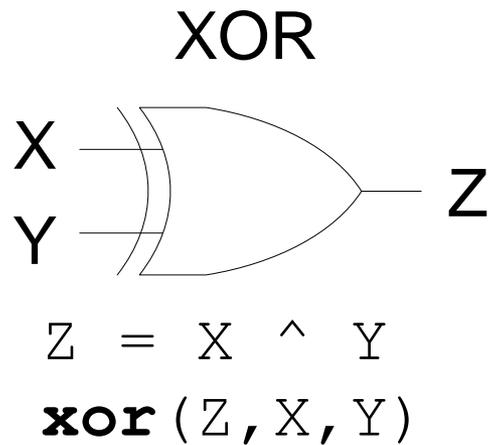


$$Z = \sim (X \mid Y)$$

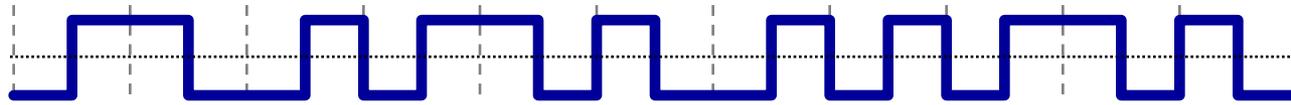
nor (Z, X, Y)

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

Exclusive-OR Gate



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

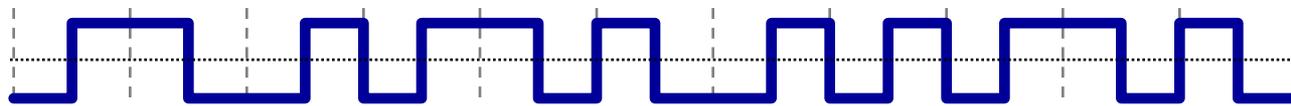


Always transition in middle of bit period:

0 = low-to-high transition

1 = high-to-low transition

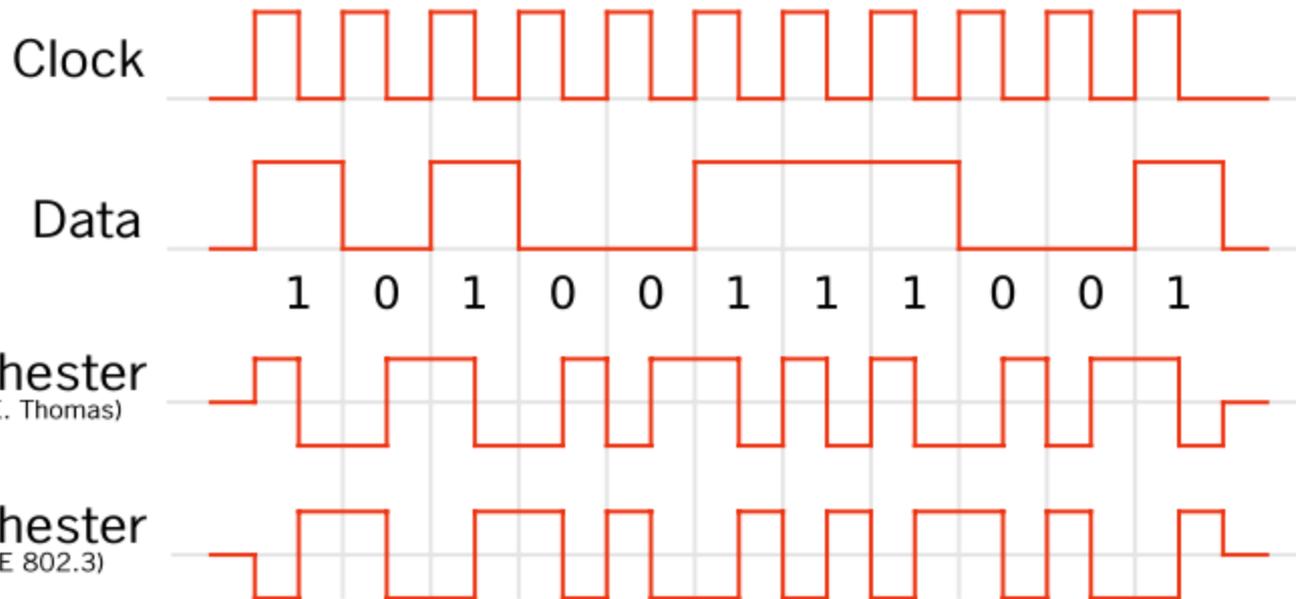
- Good clock recovery
- How to implement this?

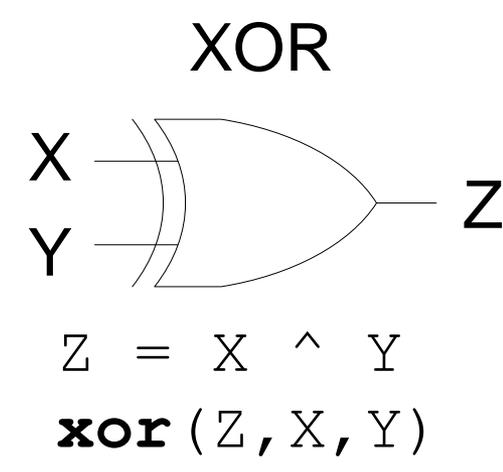
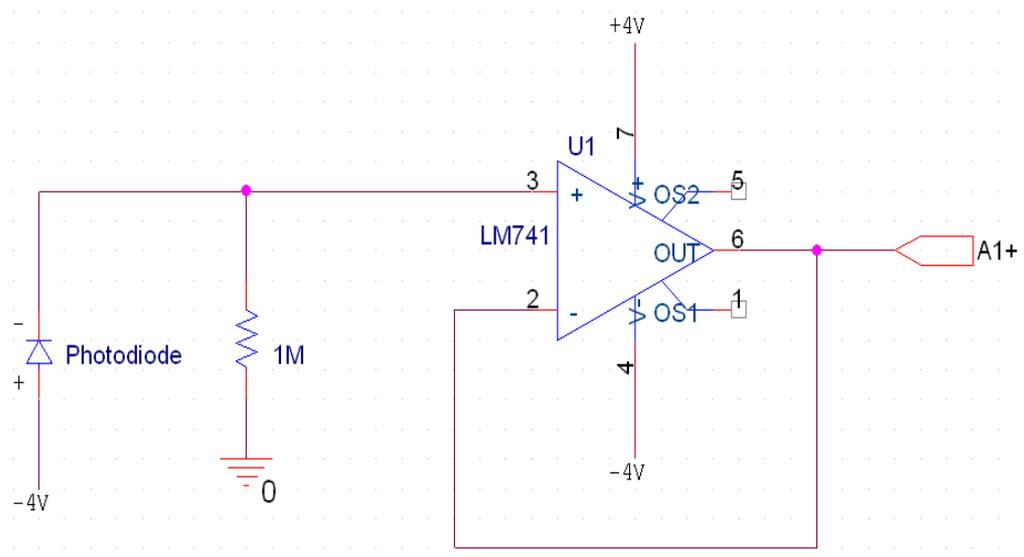


Always transition in middle of bit period:

0 = low-to-high transition

1 = high-to-low transition





X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

