Exam 2

### Boole Practice

A\*(B+C)

000 = 0

001 = 0

010 = 0

011 = 0

100 = 0

101 = 1

110 = 1

111 = 1

(A+B)\*!C

000 = 0

001 = 0

010 = 1

011 = 0

100 = 1

101 = 0

110 = 1

111 = 0

(!A+B)\*!C

000 = 1

001 = 0

010 = 1

011 = 0

100 = 0

101 = 0

110 = 1

111 = 0

!A\*B\*C

000 = 0

001 = 0

010 = 0

011 = 1

100 = 0

101 = 0

110 = 0

111 = 0

A\*!B\*!C

000 = 0

001 = 0

010 = 0

011 = 0

100 = 1

101 = 0

110 = 0

111 = 0

!A\*!B\*C

000 = 0

001 = 1

010 = 0

011 = 0

100 = 0

101 = 0

110 = 0

111 = 0

(A\*!C)+(B\*C)

000 = 0

001 = 0

010 = 0

011 = 1

100 = 1

101 = 1

110 = 1

111 = 1

(A\*!B)+(B\*C)

000 = 0

001 = 0

010 = 0

011 = 1

100 = 1

101 = 1

110 = 0

111 = 1

(A+!C)\*(!B+!C)

000 = 1

001 = 0

010 = 1

011 = 0

100 = 1

101 = 1

110 = 1

111 = 0

!A\*!B+!C

000 = 1

001 = 1

010 = 1

011 = 0

100 = 1

101 = 0

110 = 1

111 = 0

A+B\*!C

000 = 0

001 = 0

010 = 1

011 = 0

100 = 1

101 = 1

110 = 1

111 = 1

!A\*!C+!B

000 = 1

001 = 1

010 = 1

011 = 0

100 = 1

101 = 1

110 = 0

111 = 0

!A+!B\*!C

000 = 1

001 = 1

010 = 1

011 = 1

100 = 1

101 = 0

110 = 0

111 = 0

A+B\*C

000 = 0

001 = 0

010 = 0

011 = 1

100 = 1

101 = 1

110 = 1

111 = 1

!A+!B+!C

000 = 1

001 = 1

010 = 1

011 = 1

100 = 1

101 = 1

110 = 1

111 = 0

A+B+!C

000 = 1

001 = 0

010 = 1

011 = 1

100 = 1

101 = 1

110 = 1

111 = 1

(A+!B)\*(B+C)

000 = 0

001 = 1

010 = 0

011 = 0

100 = 0

101 = 1

110 = 1

111 = 1

!(A+B)+(B\*C)

000 = 1

001 = 1

010 = 0

011 = 1

100 = 0

101 = 0

110 = 0

111 = 1

A\*!(B+!C)

000 = 0

001 = 0

010 = 0

011 = 0

100 = 0

101 = 1

110 = 0

111 = 0

A+!(B+C)

000 = 1

001 = 0

010 = 0

011 = 0

100 = 1

101 = 1

110 = 1

111 = 1

!A\*B+C

000 = 0

001 = 1

010 = 1

011 = 1

100 = 0

101 = 1

110 = 0

111 = 1

A\*!C+B

000 = 0

001 = 0

010 = 1

011 = 1

100 = 1

101 = 0

110 = 1

111 = 1

(!A+B)\*(B+C)

000 = 0

001 = 1

010 = 1

011 = 1

100 = 0

101 = 0

110 = 1

111 = 1

!A\*(!B+!C)

000 = 1

001 = 1

010 = 1

011 = 0

100 = 0

101 = 0

110 = 0

111 = 0

A\*!(!B+C)

000 = 0

001 = 0

010 = 0

011 = 0

100 = 0

101 = 0

110 = 1

111 = 0

!A\*B\*!C

000 = 0

001 = 0

010 = 1

011 = 0

100 = 0

101 = 0

110 = 0

111 = 0

!(A+!C)\*!(!B+C)

000 = 0

001 = 0

010 = 0

011 = 0

100 = 0

101 = 0

110 = 0

111 = 0

!(A+!C)\*(!B+C)

000 = 0

001 = 1

010 = 0

011 = 1

100 = 0

101 = 0

110 = 0

111 = 0

A+!(!B+!C)

000 = 0

001 = 0

010 = 0

011 = 1

100 = 1

101 = 1

110 = 1

111 = 1

A\*B+!C

000 = 1

001 = 0

010 = 1

011 = 0

100 = 1

101 = 0

110 = 1

111 = 1

A\*C+B

000 = 0

001 = 0

010 = 1

011 = 1

100 = 0

101 = 1

110 = 1

111 = 1

(!A+C)\*!(B+!C)

000 = 0

001 = 1

010 = 0

011 = 0

100 = 0

101 = 1

110 = 0

111 = 0

(A+B)\*C

000 = 0

001 = 0

010 = 0

011 = 1

100 = 0

101 = 1

110 = 0

111 = 1

!A+!B+C

000 = 1

001 = 1

010 = 1

011 = 1

100 = 1

101 = 1

110 = 0

111 = 1

!A\*B+!C

000 = 1

001 = 0

010 = 1

011 = 0

100 = 1

101 = 1

110 = 1

111 = 0

(!A+!C)\*(B+C)

000 = 0

001 = 1

010 = 1

011 = 1

100 = 0

101 = 0

110 = 1

111 = 0

(!A\*!B)+(A\*C)

000 = 1

001 = 1

010 = 0

011 = 0

100 = 0

101 = 1

110 = 0

111 = 1

!A\*!B\*!C

000 = 1

001 = 0

010 = 0

011 = 0

100 = 0

101 = 0

110 = 0

111 = 0

!A+!(!B+!C)

000 = 1

001 = 1

010 = 1

011 = 1

100 = 0

101 = 0

110 = 0

111 = 1

A\*(B+!C)

000 = 0

001 = 0

010 = 0

011 = 0

100 = 1

101 = 0

110 = 1

111 = 1

!A\*(B+C)

000 = 0

001 = 1

010 = 1

011 = 1

100 = 0

101 = 0

110 = 0

111 = 0

!A\*(B+!C)

000 = 1

001 = 0

010 = 1

011 = 1

100 = 0

101 = 0

110 = 0

111 = 0

!A+B\*!C

000 = 1

001 = 1

010 = 1

011 = 1

100 = 0

101 = 0

110 = 1

111 = 0

!A+B+C

000 = 1

001 = 1

010 = 1

011 = 1

100 = 0

101 = 1

110 = 1

111 = 1

(!A+!C)\*(B+!C)

000 = 1

001 = 0

010 = 1

011 = 1

100 = 1

101 = 0

110 = 1

111 = 0

(!A\*!C)+(B\*C)

000 = 1

001 = 0

010 = 1

011 = 1

100 = 0

101 = 0

110 = 0

111 = 1

A+!B\*!C

000 = 1

001 = 0

010 = 0

011 = 0

100 = 1

101 = 1

110 = 1

111 = 1

A\*!C+!B

000 = 1

001 = 1

010 = 0

011 = 0

100 = 1

101 = 1

110 = 1

111 = 0

A\*B+C

000 = 0

001 = 1

010 = 0

011 = 1

100 = 0

101 = 1

110 = 1

111 = 1

A\*!B\*C

000 = 0

001 = 0

010 = 0

011 = 0

100 = 0

101 = 1

110 = 0

111 = 0

A+!B+C

000 = 1

001 = 1

010 = 0

011 = 1

100 = 1

101 = 1

110 = 1

111 = 1

(!A+B)\*C

000 = 0

001 = 1

010 = 0

011 = 1

100 = 0

101 = 0

110 = 0

111 = 1

(A+B)\*(B+C)

000 = 0

001 = 0

010 = 1

011 = 1

100 = 0

101 = 1

110 = 1

111 = 1

!A+B+!C

000 = 1

001 = 1

010 = 1

011 = 1

100 = 1

101 = 0

110 = 1

111 = 1

A\*!B+!C

000 = 1

001 = 0

010 = 1

011 = 0

100 = 1

101 = 1

110 = 1

111 = 0

(!A+!B)\*(A+C)

000 = 0

001 = 1

010 = 0

011 = 1

100 = 1

101 = 1

110 = 0

111 = 0

A\*B\*!C

000 = 0

001 = 0

010 = 0

011 = 0

100 = 0

101 = 0

110 = 1

111 = 0

A+!B+!C

000 = 1

001 = 1

010 = 1

011 = 0

100 = 1

101 = 1

110 = 1

111 = 1

!A\*C+B

000 = 0

001 = 1

010 = 1

011 = 1

100 = 0

101 = 0

110 = 1

111 = 1

!A+B\*C

000 = 1

001 = 1

010 = 1

011 = 1

100 = 0

101 = 0

110 = 0

111 = 1

### Boole Rules Practice

A+!A=0

Invalid

A+(B\*C)=(A+B)\*(A+C)

Distributive

A+A=A

Idempotent

!A+!B=!(A\*B)

deMorgan

A\*0=A

Invalid

A+!A=1

Inverse

A+1=1

Null

A+1=A

Invalid

A\*!A=0

Inverse

!(A\*B)=!A\*!B

Invalid

A\*0=0

Null

A\*!A=1

Invalid

!(A+B)=!A\*!B

deMorgan

(A\*B)+C=A\*(B+C)

Invalid

!A\*!B=!(A+B)

deMorgan

A\*(B+C)=(A\*B)+(A\*C)

Distributive

A\*!(B+C)=!(A\*B)+!(A\*C)

Invalid

(A\*B)+(A\*C)=A+(B\*C)

Invalid

!(A\*B)=!A+!B

deMorgan

!A\*!B=!(A\*B)

Inavlid (was distributive)

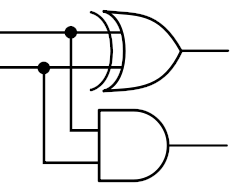
!(A+B)=!A+!B

Invalid

!A+!B=!(A+B)

Invalid

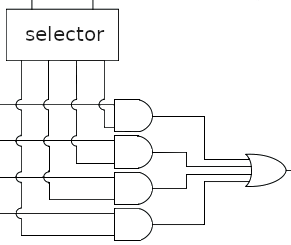
### Circuit ID



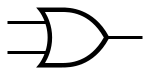
Half adder



AND



Multiplexer



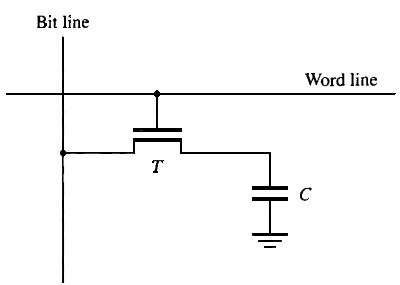
OR



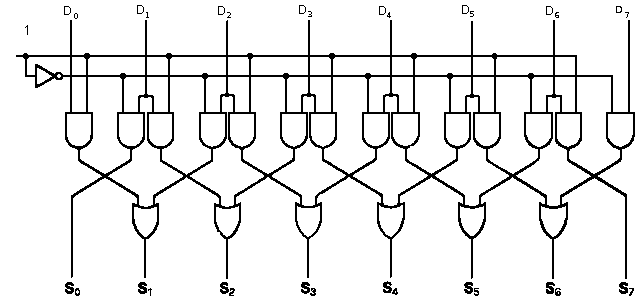
Field effect transistor



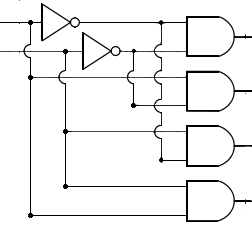
Bipolar Junction Transistor



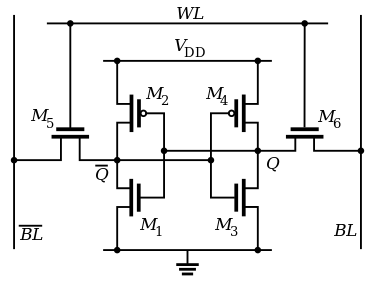
DRAM cell



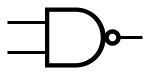
Shifter



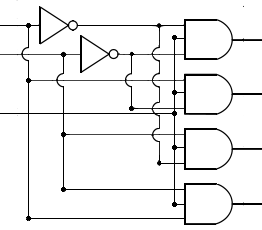
Decoder



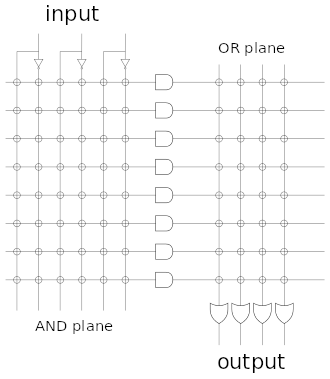
SRAM Cell



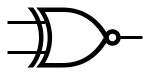
NAND



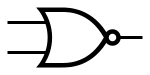
Demultiplexer



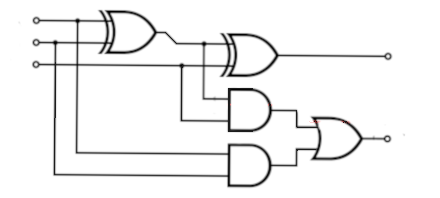
Programmable Logic Array or Array Logic



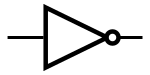
XNOR



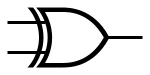
NOR



Full Adder



NOT



XOR

### ROM and RAM

|  |  |
| --- | --- |
| 1. Dynamic Ram 6. Serial EEPROM 11. SDRAM  2. Static Ram 7. Parallel EEPROM 12. Magnetic Core  3. Mask ROM 8. Flash ROM. 13. PROM  4. EPROM 9. EEPROM 14. NAND Flash ROM  5. DRAM 10. DDR SDRAM 15. NOR Flash ROM  16. Fast Page DRAM 17. EDO DRAM 18. BEDO DRAM | |

1|1\.|5|5\.|10|10\.|11|11\.

Memory that requires refresh of accessed cell but only if read.

12

Uses capacitor cells requiring regular refresh of the stored bit.

1, 5, 10, 11

Uses 6 transistor flip-flops requiring a constant power supply to hold stored bit.

2

Volatile memory that can perform 4 consecutive read requests from controller on a row of memory without resending the row address. Next cell may be addressed while current cell is being read.

17

Non-volatile memory that is read by pages and written in blocks.

14

Read/write memory that can be accessed on both the up and down tick of the clock signal.

10

Volatile memory that can perform 4 consecutive read requests from controller on a row of memory without resending the row address. Each read must complete before next cell is addressed.

16

Non-volatile memory readable at the byte level but writing is implemented by first performing a block level erasure.

15

Non-volitile memory readable and writable at the byte level, write uses a higher than working level voltage to toggle stored data.

9

"Programmed" by repeated exposing different parts of the subconductor material to different chemicals resulting in permanent preprogrammed bits.

3

Uses fuses that can be blown to set bit to "zero". Technology also used in programable logic arrays.

13

Uses either NOR gate technology which can be read like ram memory (byte addressable) or NAND gate technology which is read at block level like a hard drive.

8

**Non-volitile memory readable at the byte level but writing is implemented by first performing a block level erasure**

**15**

**Non-volitile memory that is read by pages and written in blocks.**

**14**

**Erasable using ultraviolet light.**

**4**

**Memory that requires periodic refresh of cell rows even if not being accessed.**

**1, 11, 5, 10, 16, 17, 18**

Volitile memory that can perform 4 consecutive read requests. Cell (column) addressing handled internally in the memory chip. Next cell may be addressed while current cell is being read.

18

Volitile memory that supports burst reads of 2, 4, or 8 consecutive words between refreshes.

11, 10