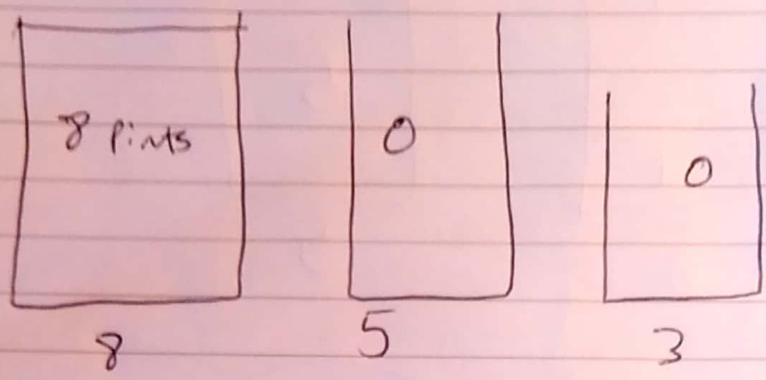
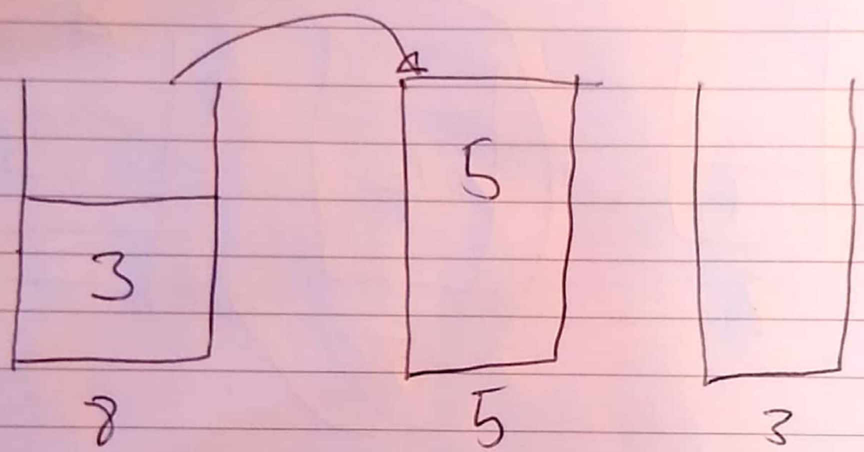
CAR

Solution credit: Die Hard 3

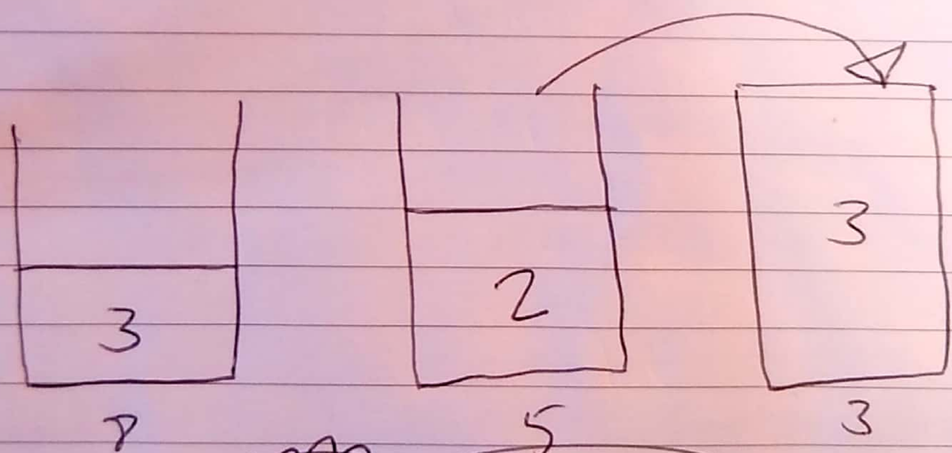
Step 0:



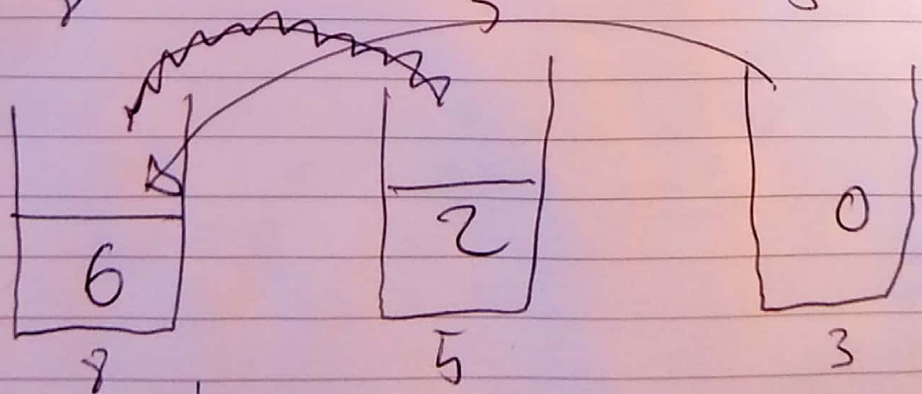
Step 1:



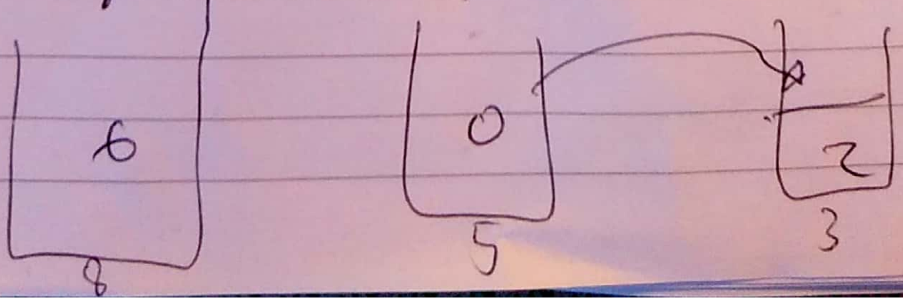
Step 2:

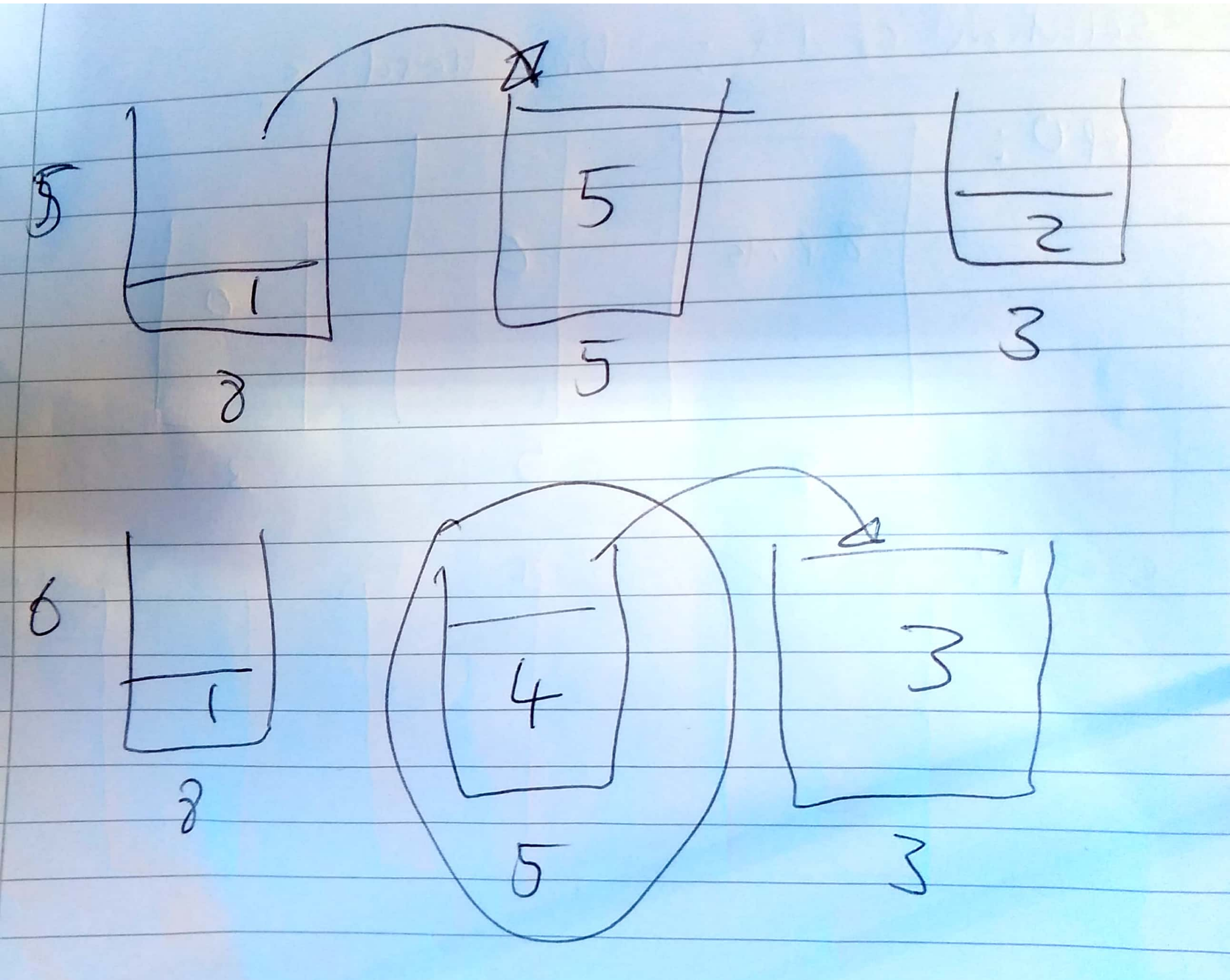


Step 3:



4




5
1
8
5
5
2
3
6
1
8
4
5
3
3

Lect: 1

Logic.Ly

Computer Architecture.

Assessment : Project based
Assessment cw Project 1 : Design a burglar alarm system
" cw Project 2 : Optimise the design.
" cw Project 3 : CPU Design/Research
" cw Project 4 : OS Design and Presentation.
Assessment 5 : Final Exam, Multiple choice, fill in the blanks.

Course Outline : Binary Arithmetic, 1's complement, 2's complement, basic gates,

Binary Addition :

0 + 0 = 0
1 + 0 = 1
0 + 1 = 1
1 + 1 = 10 ... because binary is base2

```
  1 1 0
+ 0 1 1
-------
1 0 0 1
  1
```

110 + 011 = 1001

1001 = 7
$2^3 + 0 + 0 + 2^0$

Binary Subtraction :

Basic Permutations :
0 − 0 = 0
1 − 0 = 1
1 − 1 = 0
10 − 1 = 1

```
  1 ~~1~~⁰ 10     6
− 0  0   1     − 5
---------
  1  0   1       1
```

1 0 1
$2^2 + 0 + 2^0 = 5$

Binary Multiplication:

$0 \times 0 = 0$

$1 \times 0 = 0$

$0 \times 1 = 0$

$1 \times 1 = 1$

$$
\begin{array}{r}
11 \\
\times\ 10 \\
\hline
00 \\
110 \\
\hline
110
\end{array}
\qquad
\begin{array}{l}
= 3 \\
= 2 \quad \times \\
\\
\\
\\
\hline
= 6
\end{array}
$$

$\boxed{b_{10} \rightleftharpoons b_2}$

164

Step 1: $\frac{164}{2} = 82\ r\ 0$

$\frac{82}{2} = 41\ r\ 0$

$\frac{41}{2} = 20\ r\ 1$

$\frac{20}{2} = 10\ r\ 0$

$\frac{10}{2} = 5\ r\ 0$

$\frac{5}{2} = 2\ r\ 1$

$\frac{2}{2} = 1\ r\ 0$

$\frac{1}{2} = 0\ r\ 1$

~~164~~
~~$10^{2} \cdot 10^{1} \cdot 10^{0}$~~

~~10~~
~~. 10~~
~~00~~
~~100~~
~~100 + 40~~

~~100~~
~~10~~
~~+ 1~~
~~111~~

$(164)_{10} = (10100100)_2$

~~$\frac{10110}{10} = 1011\ r\ 0$~~

~~$\frac{1011}{10} =$~~

$\frac{81}{2} = 40\ r\ 1$

$\frac{40}{2} = 20\ r\ 0$

$\frac{20}{2} = 10\ r\ 0$

$\frac{10}{2} = 5\ r\ 0$

$\frac{5}{2} = 2\ r\ 1$

$\frac{2}{2} = 1\ r\ 0$

$\frac{1}{2} = 0\ r\ 1$

$(81)_{10} = (1010001)_2$

Binary Division

```
       1 1
10 ) 1 1 0
    -1 0 ↓
     0 1 0
     - 1 0
       0 0
```

$$\frac{110}{10} = 11$$

```
         0 0 0
11 ) 1 0 0 1 0
    -1 1 ↓
     0 1 0 1 0
      -1 1 ↓ ↓
       0 1 0 0
      -1 1
```

2's Complement

0	0	0	0	0	0	0	0
128	64	32	16	8	4	2	1

0 0 0 1 0 1 0 0
= 20

"Keep every thing until the first RHS '1' value and flip the rest"

2's complement 1 1 1 0 1 1 0 0
= -20

$$\begin{array}{r} 24 \\ +(-20) \\ \hline 4 \end{array} \qquad \begin{array}{r} 0\,0\,0\,1\,1\,0\,0\,0 \\ +\ 1\,1\,1\,0\,1\,1\,0\,0 \\ \hline \cancel{1}\ 0\,0\,0\,0\,0\,1\,0\,0 \end{array} = \boxed{4}$$

cut of after 8 bits.

1's complement

0 → 1
1 → 0

WEEK 2

CAR

NEGATIVE NUMBERS

CAR Work Schedule live on Moodle

```
   1  0  0  1  0
 - 1  0  1  1  1
 ---------------
   0  0  0  1  1
```

Complement

1's - reverse the bits.

```
   1 1 0 1 0            1 1 0 1 0
 - 1 0 1 1 1    ->    + 0 1 0 0 0
 -----------          -----------
   0 0 0 1 1          1 0 0 0 1 0
                              + 1
                      -----------
                        0 0 0 1 1
```

```
                 1's
 - 0 1 1 1 0     ->     1 0 0 0 1

                        1 1 0 0 0
                      + 1 0 0 0 1
                      -----------
                        0 1 0 0 1
                              + 1
                      -----------
                        0 1 0 1 0
```

2's complement

1's + 1

$$\begin{array}{r} 11000 \\ -\,01110 \\ \hline \end{array}$$

$-01110 \xrightarrow{2's} 10001$

$$\begin{array}{r} 10001 \\ +\quad 1 \\ \hline 10010 \end{array}$$

$$\begin{array}{r} 11000 \\ -\,\cancel{+}\,01110 \\ \hline \end{array} \xrightarrow{2's} \begin{array}{r} 11000 \\ +\,10010 \\ \hline \cancel{1}\,\boxed{01010} \end{array}$$

CAR: Sheet 1: Binary Arithmetic

Core Exercises

1) a)

```
  11
+ 11
 110
```

11 + 11 = 110

b)

```
  100
+ 010
  110
```

100 + 10 = 110

c)

```
  111
+ 011
 1010
```

111 + 11 = 1010

d)

```
  110
+ 100
 1010
```

110 + 100 = 1010

e)

```
  1111
+ 1100
 11011
```

1111 + 1100 = 11011

f)

```
  11
- 01
  10
```

11 − 01 = 10

g)

```
  11
- 10
  01
```

11 − 10 = 01

h) 111 − 100 =

$$\begin{array}{r} 111 \\ -100 \\ \hline 011 \end{array}$$

$= \boxed{011}$ ✓

~~k)~~

$2^2 - (2^2 + 2^1 + 2^0) = 4 - 7 = -3$

~~100 − 111 = 011~~

i)

$$\boxed{\begin{array}{r} 101 \\ -011 \\ \hline 010 \end{array}}$$

$\boxed{101 - 011 = 010}$ ✓

j)

$$\begin{array}{r} 11 \\ \times 11 \\ \hline 11 \\ +110 \\ \hline 1001 \end{array}$$

11 × 11 = 1001 ✓

k)

$$\begin{array}{r} 101 \\ \times 111 \\ \hline 101 \\ 1010 \\ 10100 \\ \hline 100011 \end{array}$$

101 × 111 = 100011 ✓

l)

$$\begin{array}{r} 1101 \\ \times 1010 \\ \hline 0000 \\ 11010 \\ 000000 \\ +1101000 \\ \hline 10000010 \end{array}$$

1101 × 1010 = 10000010 ✓

m)

```
      1 0
11 | 1 1 0
     1 0
     0 0
```

$\frac{110}{11} = 10$ ✓

n)

```
~~10 | 1 1 0~~
~~     1 0 0~~
~~     0 1 0~~
```

```
     1 1
10 | 1 1 0
   - 1 0 0
     0 1 0
   -   1 0
       0 0
```

$\frac{110}{10} = 11$ ✓

o)

```
       0 0 1 1
100 | 1 1 0 0
    - 1 0 0 0
      0 1 0 0
    -   1 0 0
        0 0 0
```

$\frac{1100}{100} = 11$

2)

a) 11 + 01 = 100 ✓

b) 10 + 10 = 100 ✓

c) 101 + 11 = 1000 ✓

d) 111 + 110 = 1001 × 1101

e) 1001 + 101 = 1110 ✓

f) 1101 + 1011 = 11000

g) $\begin{array}{r} 11 \\ -01 \\ \hline 10 \end{array}$ $11 - \cancel{0}1 = 10$ ✓

h) $101 - 100 = 001$

i) $1\overset{0}{\cancel{1}}{}^{1}0 - 101 = 001$

j) $1\overset{0}{\cancel{1}}{}^{1}\overset{0}{\cancel{1}}{}^{1}0 - 11 = 1011$

k) $1\overset{0}{\cancel{1}}\overset{0}{\cancel{0}}\overset{0}{0} - 1001 = 00\cancel{0}1$

l) ~~$1\overset{0}{\cancel{1}}\overset{0}{\cancel{0}}\overset{1}{\cancel{1}}\overset{1}{\cancel{1}}0 - 10111 = 11$~~

$\begin{array}{r} 1\overset{0}{\cancel{1}}\,\overset{0}{\cancel{0}}\,\overset{1}{\cancel{1}}\,{}^{1}0 \\ -\ 10111 \\ \hline 00011 \end{array}$ $11010 - 10111 = 00011$

=

m) $\begin{array}{r} 11 \\ \times\ 11 \\ \hline 11 \\ 110 \\ \hline 1001 \end{array}$ $11 \times 11 = 1001$ ✓

n) $\begin{array}{r} 100 \\ \times\ 010 \\ \hline 1000 \end{array}$ $100 \times 10 = 1000$

o) 111 × 101 = 100011

```
   111
 × 101
 -----
   111
 11100
------
100011
```

p) 1001 × 110 = 110110

```
   1001
 ×  110
 ------
  10010
 100100
 ------
 110110
```

q) 1101 × 1101 = 10101001

```
     1101
   × 1101
   ------
     1101
   110100
  1101000
 --------
 10101001
```

r) 1110 × 1101 = 10110110

```
     1110
   × 1101
   ------
     1110
   111000
  1110000
 --------
 10110110
```

s) $\frac{100}{10} = 10$ ✓

t)

```
        0 0 1 1
11 | 1 0 0 0 1
   -  0 1 1 0
      1 0 1 1
        - 1 1 1
            0
```

$\frac{1001}{11} = 0011$ ✓

u)

```
         0 0 1 1
100 | 1 1 0 0
    - 1 0 0 ↓
      0 1 0 0
      - 1 0 0
        0 0 0
```

$\frac{1100}{100} = 11$

Extension Exercises

3) Multiply two binary numbers using only binary addition.

$\frac{12}{2} = 6 \, r \, 0$

$\frac{6}{2} = 3 \, r \, 0$

$\frac{3}{2} = 1 \, r \, 1$

$\frac{1}{2} = 0 \, r \, 1$

(6) (2) (12)

110 x 10 = 1100

Sum 110 for 10 number of times.

Convert one element of Multiplication to a b_{10} Number and Sum the Next Number by the amount of times provided by the b_{10} number

$$110 \times 10 = \begin{array}{r} 110 \\ +\ 110 \\ \hline \boxed{1100} \end{array}$$

$$6 \times 2$$

4) <u>Divide</u> two numbers using only binary addition and subtraction.

$$110 \div 10$$

$$\begin{array}{r} 110 \\ -\ 10 \\ \hline 100 \end{array} \qquad \begin{array}{r} \cancel{1}00 \\ -\ 10 \\ \hline 10 \end{array} \qquad \begin{array}{r} 10 \\ -\ 10 \\ \hline 0 \end{array}$$

1 2 3

~~...~~ 10 goes into 110 3 times

$$(3)_{10} = (11)_2$$

Extension Ex $1 \rightarrow 3$

1) $\frac{198}{6} = 33 \; r \; 0$ $\qquad (198)_{10} = (530)_6$

$\frac{33}{6} = 5 \; r \; 3$ ✓

$\frac{5}{6} = 0 \; r \; 5$

2) $\frac{11236}{10} = 1123 \; r \; 6$

$\frac{1123}{10} = 112 \; r \; 3$

$\frac{112}{10} = 11 \; r \; 1$ ✗

$\frac{11}{10} = 1 \; r \; 1$

$\frac{1}{10} = 0 \; r \; 1$

$(11236)_7 = (11136)_{10}$

3) $(40013)_5 \rightarrow (\;)_6$

$\frac{40013}{6} = 6668 \; r \; 5$

$\frac{6668}{6} = 1111 \; r \; 2$

$\frac{1111}{6} = 185 \; r \; 1$

$\frac{185}{6} = 30 \; r \; 5$

$\frac{30}{6} = 5 \; r \; 0$

$\frac{5}{6} = 0 \; r \; 5$ ✗

Another method for converting $b_{10} \to b_2$

remainder 35 × ~~...~~ times fitting into each 2^x

remainder	35	×
35	32	1
3	16	0
3	8	0
3	4	0
3	2	1
1	1	1

CAR Sheet 1 Base Exercises

128
64
32
16
8
4
2

1) a) $\frac{12}{2} = 6$ r 0

$\frac{6}{2} = 3$ r 0

$\frac{3}{2} = 1$ r 1

$\frac{1}{2} = 0$ r 1

$\boxed{12 = 1100}$

b)

	103	
	~~128~~	
103	64	1
39	32	1
7	16	0
9	8	1
1	4	0
3	2	1

$\boxed{103 = 1100111}$

$\frac{103}{2} = 51$ r 1

$\frac{51}{2} = 25$ r 1

$\frac{25}{2} = 12$ r 1

$\frac{12}{2} = 6$ r 0

$\frac{6}{2} = 3$ r 0

$\frac{3}{2} = 1$ r 1

$\frac{1}{2} = 0$ r 1

c) $\frac{97}{2} = 48 \text{ r } 1$

$\boxed{97 = 1000001}$

$\frac{48}{2} = 24 \text{ r } 0$

$\frac{24}{2} = 12 \text{ r } 0$

$\frac{12}{2} = 6 \text{ r } 0$

$\frac{6}{2} = 3 \text{ r } 0$

$\frac{3}{2} = 1 \text{ r } \cancel{0}$

$\frac{1}{2} = 0 \text{ r } 1$

d) $\frac{55}{2} = 27 \text{ r } 1$

$\boxed{55 = 110111}$

$\frac{27}{2} = 13 \text{ r } \cancel{0}$

$\frac{13}{2} = 6 \text{ r } 1$

$\frac{6}{2} = 3 \text{ r } 0$

$\frac{3}{2} = 1 \text{ r } \cancel{0}$

$\frac{1}{2} = 0 \text{ r } 1$

e) $\frac{395}{2} = 197 \text{ r } 1$

$\boxed{395 = 110001011}$

$\frac{197}{2} = 98 \text{ r } 1$

$\frac{98}{2} = 49 \text{ r } 0$

$\frac{49}{2} = 24 \text{ r } 1$

$\frac{24}{2} = 12 \text{ r } 0$

$\frac{12}{2} = 6 \text{ r } 0$

$\frac{6}{2} = 3 \text{ r } 0$

$\frac{3}{2} = 1 \text{ r } 1$

$\frac{1}{2} = 0 \text{ r } 1$

2) a) 1 1 0 1

$2^3 + 2^2 + 2^0 = 8 + 4 + 1 = 13$ ✓

b) 1 0 1 0 0 1

$2^5 + 2^3 + 2^0 = 32 + 8 + 1 = 41$ ✓

c) 1 1 0 1 1 1

$2^5 + 2^4 + 2^2 + 2^1 + 2^0 = 32 + 16 + 4 + 2 + 1 = 55$

d) 1 0 0 0 0 1 1 1 ✓

$2^7 + 2^2 + 2^1 + 2^0 = 135$ ✓

e) 1 1 1 1 1 1 1 0

$2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 =$ $=$

$128 + 64 + 32 + 16 + 8 + 4 + 2 = 254$ ✓

1

CAR: Sheet 2: Binary Arithmetic w/ Negative Numbers.

1's comp: 0 → 1, 1 → 0

$\boxed{01101 \rightarrow 10010}$

2's comp: 1's comp + 1

011010 → 100101
+ 1
100110

~~011010 → 100101~~
~~+ 1~~
~~100110~~

01101 → 10010
+ 1
10011

$\frac{40}{2} = 20 \text{ r } 0$

$\frac{20}{2} = 10 \text{ r } 0$

$\frac{10}{2} = 5 \text{ r } 0$

$\frac{5}{2} = 2 \text{ r } 1$

$\frac{2}{2} = 1 \text{ r } 0$

$\frac{1}{2} = 0 \text{ r } 1$

Core Exercises

1's comp

1) a) 01 → 10
b) 10 → 01
c) 111 → 000
d) 011 → 100
e) 100 → 011
f) 101 → 010
g) 0011 → 1100
h) 1001 → 0110
i) 10111 → 01000

2's comp

2) a) 01 → 11
b) 10 → 10
c) 111 → 001
d) 011 → 101
e) 100 → 100
f) 101 → 011
g) 0011 → 1101
h) 1001 → 0111
i) 10111 → 01001

3) a) i) $11 - 01 \xrightarrow{1's} 00 + 10 = 10$

~~00 + 10~~

ii) $11 - 01 \xrightarrow{2's} \cancel{01 + 11} = \cancel{\begin{array}{r} 01 \\ +\ 11 \\ \hline X\ 01 \end{array}}$

$\cancel{= 01}$

$11 + 11 = \begin{array}{r} 11 \\ +\ 11 \\ \hline X\ 10 \end{array}$

b) i) $11 - 10 \xrightarrow{1's} 11 + 01 = \begin{array}{r} 11 \\ +\ 01 \\ \hline 1\ 00 \\ +\ 1 \\ \hline \boxed{01} \end{array}$

ii) $11 - 10 \xrightarrow{2's} 11 + 10$

$= \begin{array}{r} 11 \\ +\ 10 \\ \hline X\ 01 \end{array}$

c) i) $\overset{4}{100} - \overset{6}{110} \xrightarrow{1's} 100 + 001 = \begin{array}{r} 100 \\ +\ 001 \\ \hline 101 \end{array}$

old question.

ii) $100 - 110 \xrightarrow{2's} 100 + 010 = \begin{array}{r} 100 \\ +\ 010 \\ \hline 110 \end{array}$

d) i) $101 - 110 \xrightarrow{1's} 101 + 001 = \begin{array}{r} 101 \\ +\ 001 \\ \hline 110 \end{array}$

ii) $101 - 110 \xrightarrow{2's} 101 + 010$

$= 101$

$+ 1$

d) i) $101 - 011 \xrightarrow{1's} 101 + 100 =$

$$\begin{array}{r} 101 \\ +\ 100 \\ \hline 1\,101 \\ 1 \\ \hline \boxed{110} \end{array}$$

ii) $101 - 011 \xrightarrow{2's} 101 + 101$

$$\begin{array}{r} 101 \\ +\ 101 \\ \hline \cancel{1}\,110 \end{array}$$

e) i) see question 3) a)
ii)

f) i) $101 - 100 \xrightarrow{1's} 101 + 011$

$$\begin{array}{r} 101 \\ +\ 011 \\ \hline 1\,000 \\ +\ \ 1 \\ \hline 001 \end{array}$$

ii) $101 - 100 \xrightarrow{2's} 101 + 100$

$$\begin{array}{r} 101 \\ +\ 100 \\ \hline \cancel{1}\,001 \end{array}$$

g) i) $110 - 101 \xrightarrow{1's} 110 + 010$

$$\begin{array}{r} 110 \\ 010 \\ \hline 1\,000 \\ 1 \\ \hline 001 \end{array}$$

ii) $110 - 101 \xrightarrow{2's} 110 + 011$

$$\begin{array}{r} 110 \\ +\ 011 \\ \hline \cancel{1}\,001 \end{array}$$

h) i) $1110 - 0011 \xrightarrow{1's} 1110 + 1100 =$

$$\begin{array}{r} 1110 \\ +\ 1100 \\ \hline 1010 \\ 1 \\ \hline 1011 \end{array}$$

ii) $1110 - 0011 \xrightarrow{2's} 1110 + 1101$

$$\begin{array}{r} = 1110 \\ +\ 1101 \\ \hline \cancel{1}\,1011 \end{array}$$

Scanned by CamScanner

i) i) 1100 − 1001 $\xrightarrow{1's}$ 1100 + 0110 =

$$\begin{array}{r} 1100 \\ +\ 0110 \\ \hline 0010 \\ 1 \\ \hline 0011 \end{array}$$

ii) 1100 − 1001 $\xrightarrow{2's}$ 1100 + 0111

$$\begin{array}{r} = 1100 \\ +\ 0111 \\ \hline \cancel{1}0011 \end{array}$$

j) i) 11010 − 10111 $\xrightarrow{1's}$ 11010 + 01000

$$\begin{array}{r} 11010 \\ +\ 01000 \\ \hline 00010 \\ 1 \\ \hline 00011 \end{array}$$

ii) 11010 − 10111 $\xrightarrow{2's}$ 11010 + 01001

$$\begin{array}{r} 11010 \\ 01001 \\ \hline \cancel{1}00011 \end{array}$$

Extension Exercises

4) 110 ÷ 10

110 + 10 = $\begin{array}{r} 110 \\ -\ 10 \\ \hline 100 \end{array}$

$\dfrac{110}{10} = 11$

100 + 10

$$\begin{array}{r} = 0100 \\ 10 \\ \hline 10 \end{array}$$

10 + 10 = 0

5) Find 2's complement of 10 and add it to 110 until 0

$$\begin{array}{r} 010 \\ 1 \\ \hline 011 \end{array}$$

3) c) i) $101 - 011 \xrightarrow{1's} 101 + 100$

$$\begin{array}{r} 101 \\ +100 \\ \hline 001 \\ + \quad 1 \\ \hline 010 \end{array}$$

ii) $101 - 011 \xrightarrow{2's} 101 + 101$,

$$\begin{array}{r} 101 \\ +101 \\ \hline \cancel{1}\boxed{010} \end{array}$$

3) e) i) $110 - 101 \xrightarrow{1's}$

$$\begin{array}{r} 110 \\ +010 \\ \hline 000 \\ + \quad 1 \\ \hline 001 \end{array}$$

ii) $110 - 101 \xrightarrow{2's}$

$$\begin{array}{r} 110 \\ +011 \\ \hline \cancel{1}001 \end{array}$$

Extension Exercises

byte ← 1−1

4) Representations of zero

$1-1 \xrightarrow{1's}$

$$\begin{array}{r} 00000001 \\ +00000000 \\ \hline 00000001 \end{array}$$

2's does not allow for a negative rep of zero.

$1-1 \xrightarrow{2's}$

$$\begin{array}{r} 00000001 \\ +00000001 \\ \hline 00000010 \end{array}$$

Test workings

3) 10100 $\xrightarrow{2's}$ 01011

$$\begin{array}{r} 01011 \\ +\quad 1 \\ \hline 01100 \end{array}$$

$$\begin{array}{r} 01011 \\ +\quad 1 \\ \hline 01100 \end{array}$$

0
2
4
8
16

4) 11 − 01 $\xrightarrow{1's}$ 11 + 10 →

$$\begin{array}{r} 11 \\ +\ 10 \\ \hline 01 \\ 1 \\ \hline 10 \end{array}$$

5) 1110 − 0011 $\xrightarrow{1's}$ 1110 + 1100

$$\begin{array}{r} 1110 \\ +\ 1100 \\ \hline 1010 \\ 1 \\ \hline 1011 \end{array}$$

14 − 3 = 11

6) 11 − 01 $\xrightarrow{2's}$ 11 + 11

$$\begin{array}{r} 11 \\ +\ 11 \\ \hline \cancel{1}\,10 \end{array}$$

7) 10111 − 10001 $\xrightarrow{2's}$ 10111 + 0111$\cancel{0}$

$$\begin{array}{r} 10111 \\ +\ 01111 \\ \hline \cancel{1}\ 00110 \end{array}$$

$$\begin{array}{r} \cancel{10111} \\ +\ \cancel{00110} \\ \hline \cancel{1\,00101} \end{array}$$

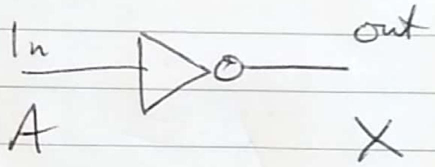
WEEK 3

CAR

- Course work
- Logic - Ly
- Gates.

- link together 3 burglar alarm Systems that use logic gates.
- Hopefully integrated subsystems.
- Demonstration assessment
  - ↳ Why we did what we did.
  - ↳ Explaining process and understanding of specification.
- Week to descipher the specification.

NOT - inverter.

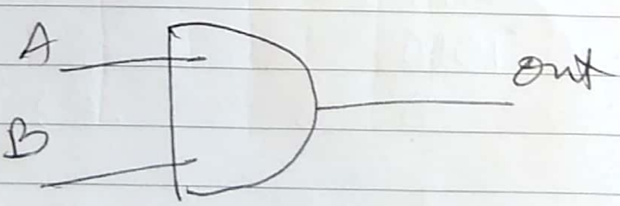


In	Out
1	0
0	1

Notation: $X = A'$ Bar/Apostrophe, makes X, output, equal to inversion of A, input.

$X = \overline{A}$

AND



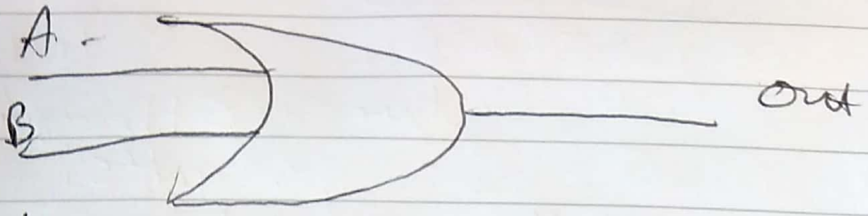
A	B	out
1	1	1
1	0	0
0	1	0
0	0	0

Notation:

$X = AB = A.B = A \wedge B$

And gate multiplies the input.

OR



A	B	Out
1	1	1
0	0	0
1	0	1
0	1	1

Notation!

$$X = A + B = A \vee B$$

$$X = (A+B)\,C = (A \vee B) \wedge C$$

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

WK 4

CAR:

UNIVERSAL GATES

↳ can be used to build any circuit.

~~Set~~ Course work:

- Set up each circuit individually initially

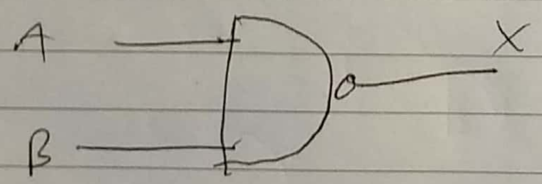
- Test each circuit against scenario
  ↳ Truth tables

Push button – sensor
Toggle switch – switch.
Light ~~bulb~~ can represent any required output – label it.

Circuit integration ~~or~~ required for higher marks.

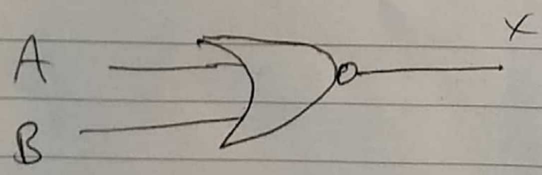
NAND / NOR Notation

NAND



$X = (AB)'$

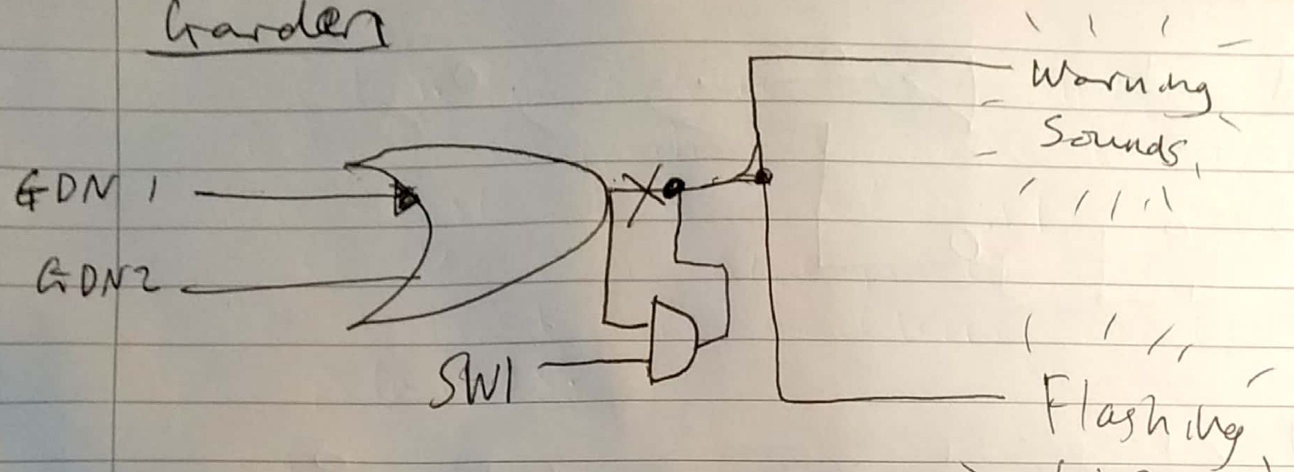
NOR



$X = (A+B)'$

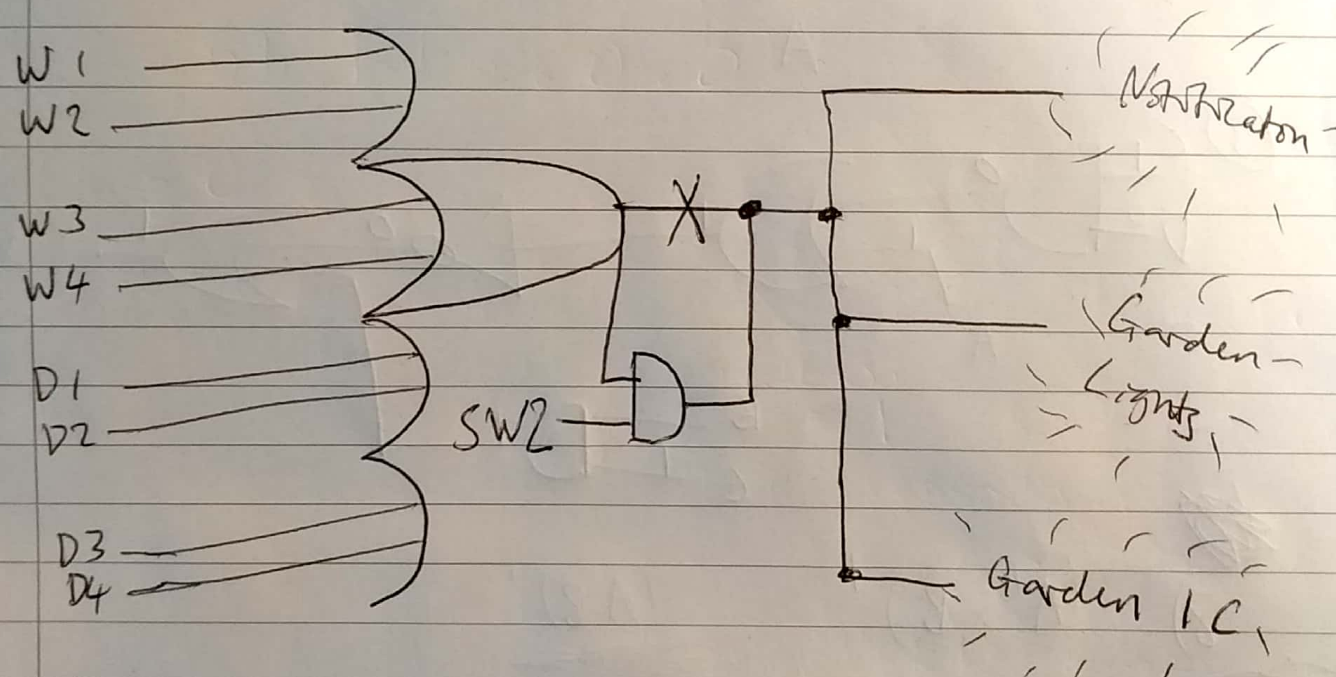
CAR COURSEWORK NOTES

Garden

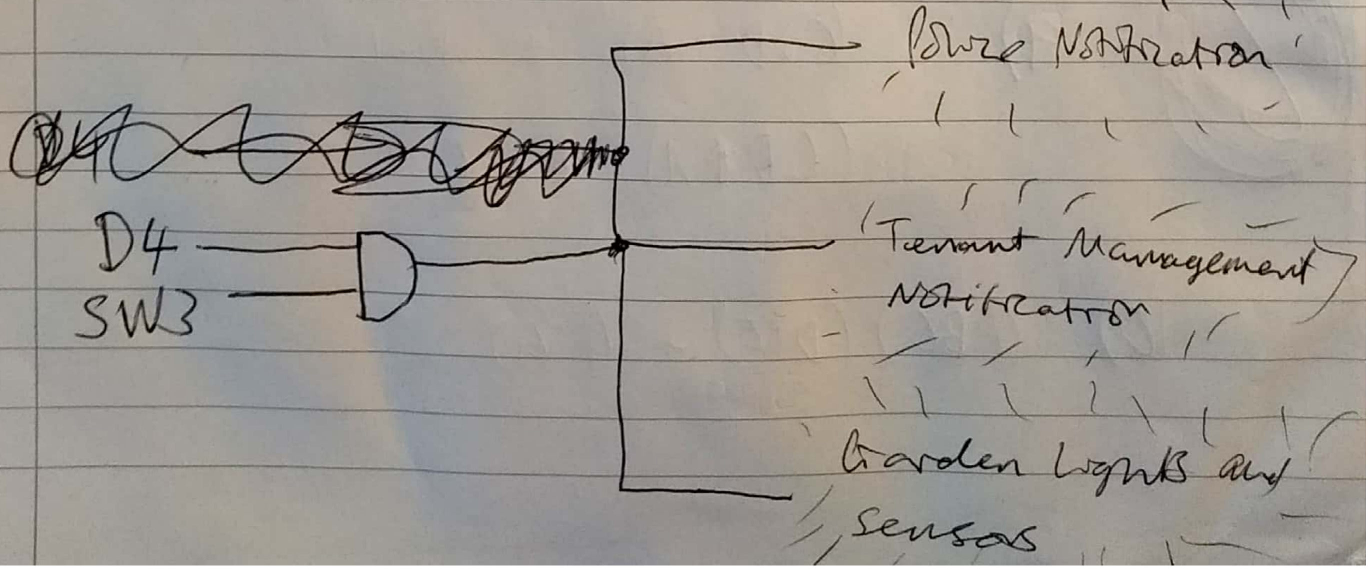


House

Doors and windows prefaced with a NOT gate.



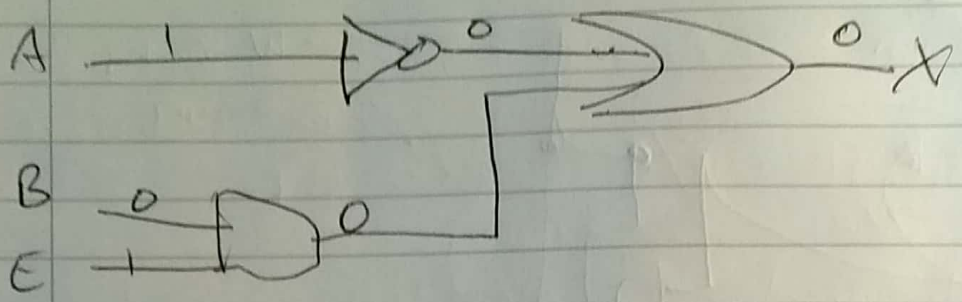
Safe



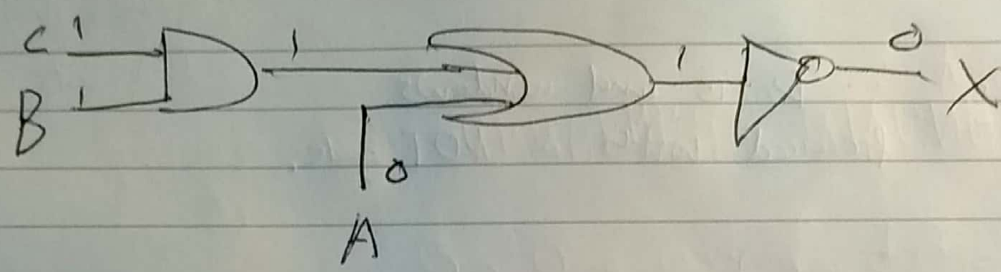
Scanned by CamScanner

CAR Test Notes: NOT, AND, OR

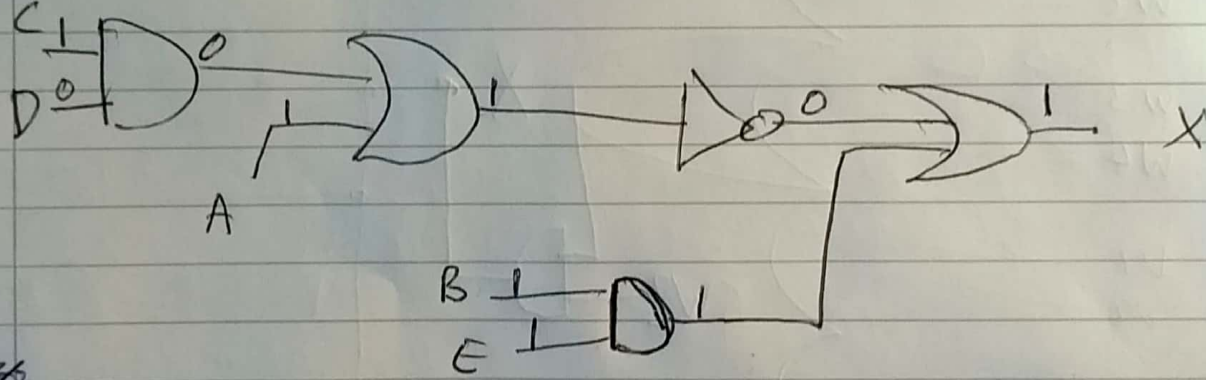
①



②



③



8) $(A' \times B)' = (A'B)'$

9) $(C \times D + A)' + B \times E$

$(CD + A)' + BE$

10) $(ABC)(D+E) + (FG)$

Wk 5

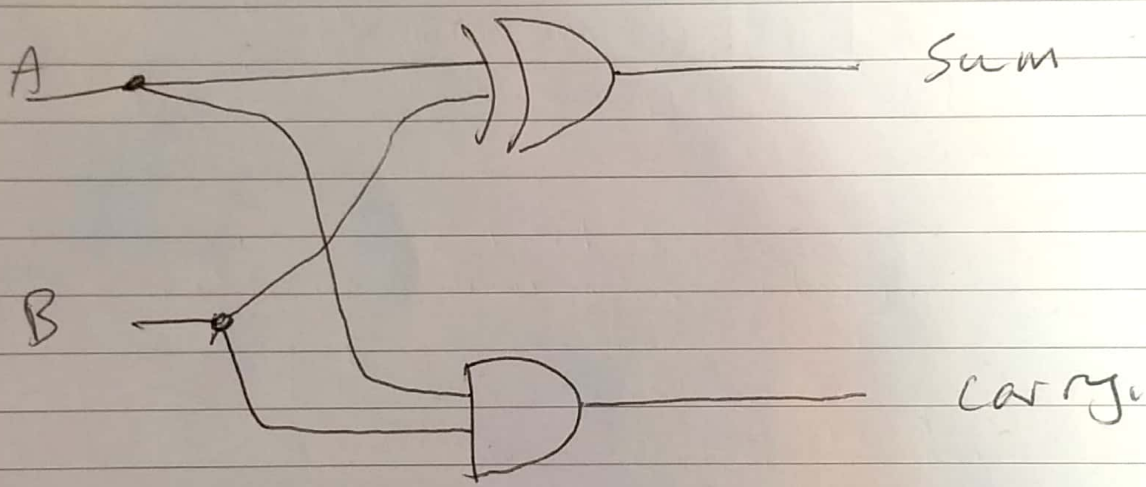
CAR

Sum + carry w/ gates

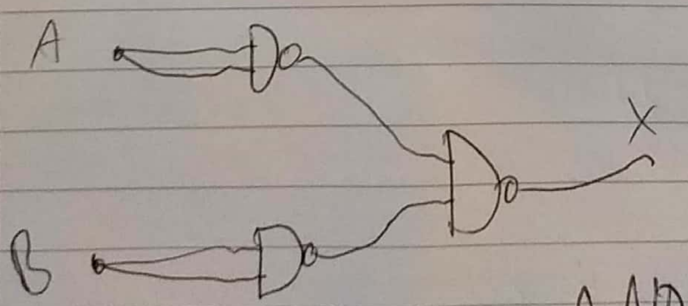
$$\begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \end{array}$$

carry sum.

Half Adder



A	B	Sum	Carry
1	1	0	1
0	0	0	0
1	0	1	0
0	1	1	0

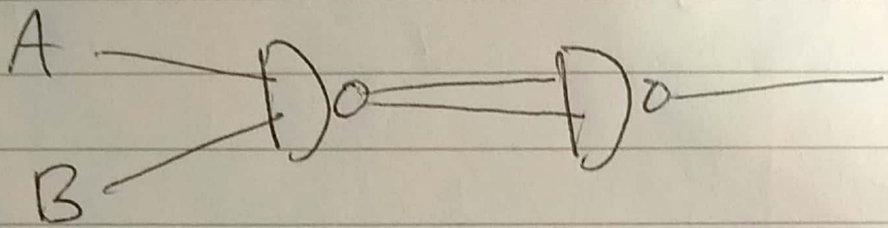


OR from AND

~~AN~~

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

AND from NAND



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

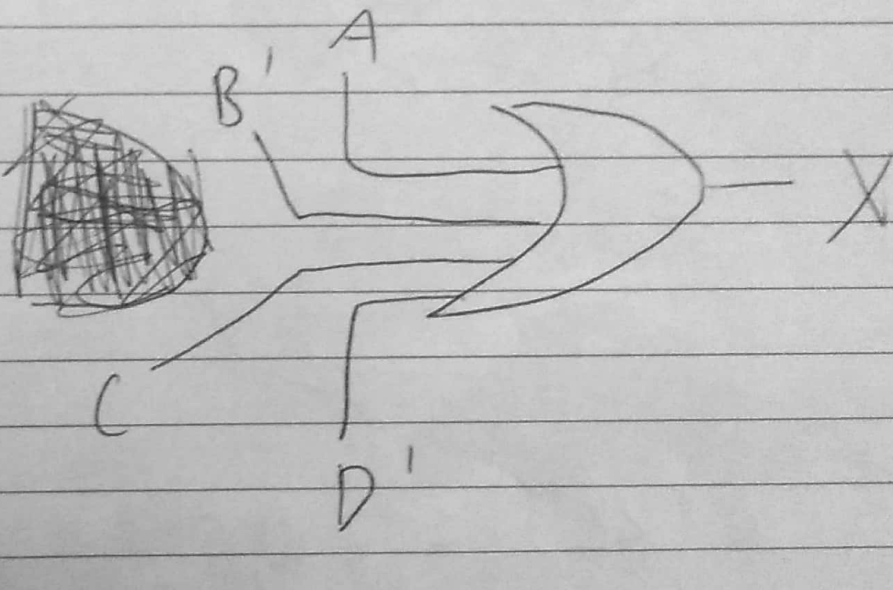
Wk 7

CAR

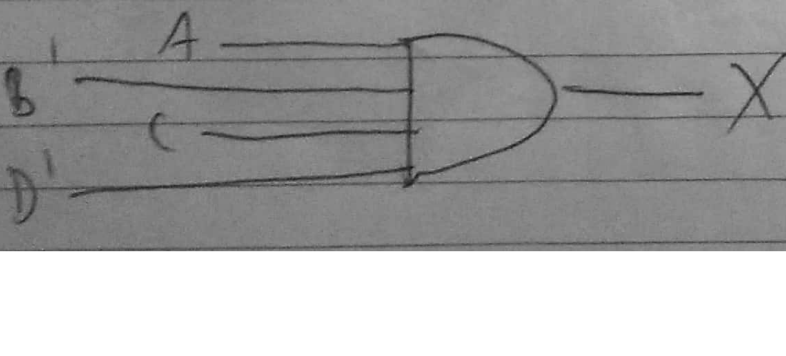
Second Assignment - Optimisation.
- Universal Gate Version.
- Using boolean Algebra.
- Karnaugh Maps.

Boolean Algebra

$$X = A + B' + C + D'$$



$$X = AB'CD'$$

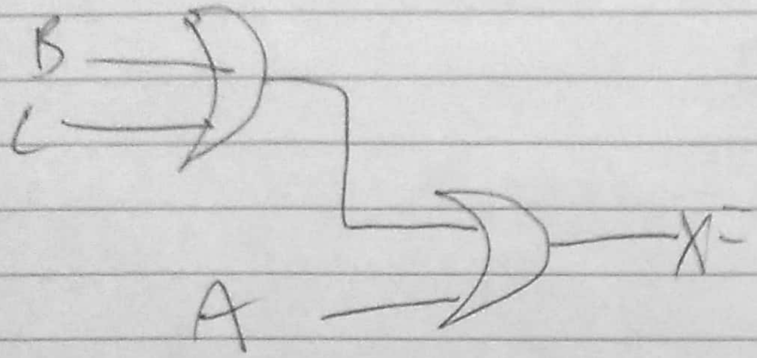


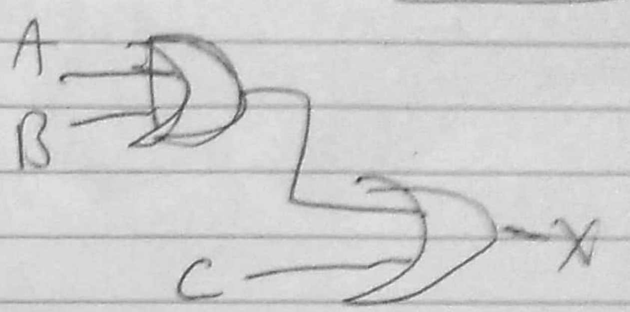
Boolean Algebra Laws

$A + B = B + A$ * Commutative Law *

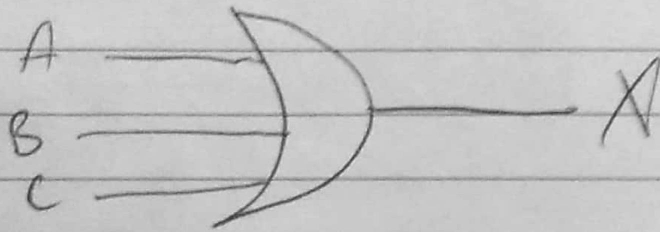
$AB = BA$ *

$A+B+C = A + (B+C) = (A+B) + C$ * Associative LAW ~~Rule~~ *



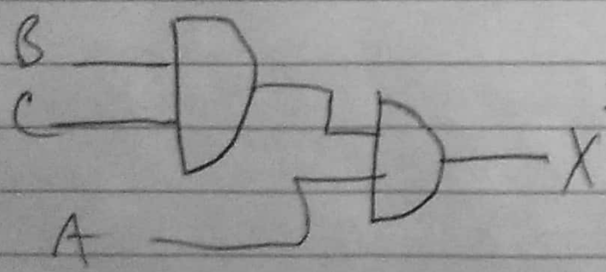


=

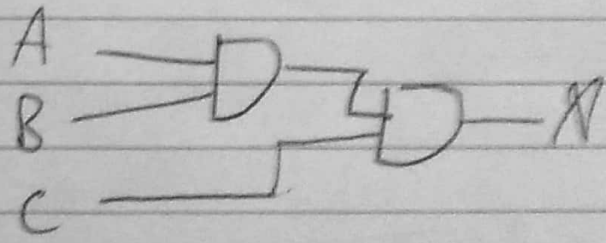


≡

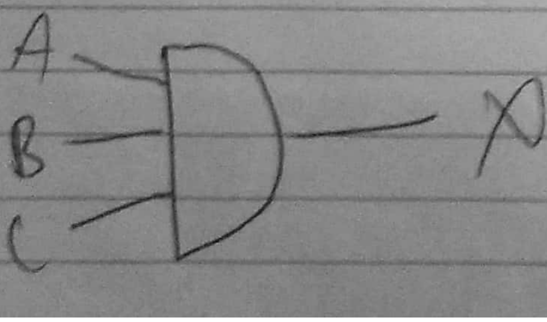
$A(BC) = (AB)C$



≡



≡



wk 8 CAR

Have a read:

9
- standard SOP form
- Karnaugh maps.

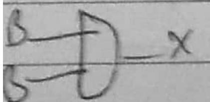
8
- Laws and Rules
- DeMorgan's Theorems
- Simplification

$$X = AB + A(B+C) + B(C+B)$$

distributive law.

$$X = AB + AB + AC + BB + BC$$

$$X = AB + AB + AC + \boxed{BC + B} = B$$



B	B	X
1	1	1
0	0	0

B	X
1	1
0	0

$$X = A(B+B+C) + \cancel{B(C+1)}$$

$$X = A(B+C) + BC + B$$

$$X = AB + AC + BC + B$$

$$\cancel{X = A(B+C) + B(C+1)}$$

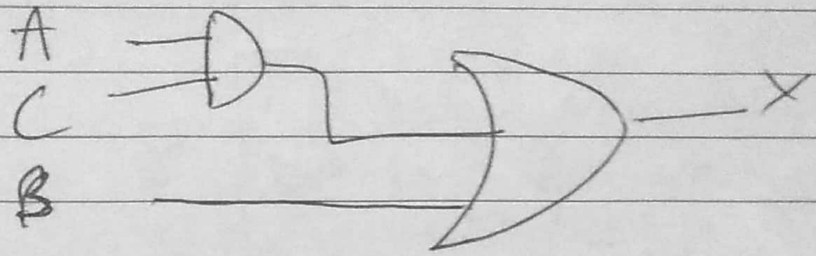
$$X = C(A+B) + B(A+1)$$

$$= A(B+C) + B(C+1)$$

$$X = \underbrace{\boxed{AB + AB}}_{= AB} + AC + B$$

$X = \underline{AB} + AC + \underline{B}$

$$\boxed{X = \underline{AC + B}}$$



$X = C(A'.B' + (AB').(C + D.B))$

$$\boxed{X = C(A'B' + (AB').(C + B.D))}$$

$= CA'B' + C(AB').(C + B.D)$

$= CA'B' + CAB'.(C + B.D)$

$= \boxed{B'C}$

$XX = X$

$X'X = 0$

Wk 9

CAR - Optimization

CW2

- Look at the prioritization of the marking criteria categories.

Minimize the number of gates.

- Sum of products form and Karnaugh maps.

Karnaugh Maps

- Only need to think about the combinations where Output = 1

$(CD + B)A$

$A = 1$

$CD + B = 1$

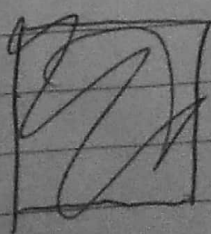
A	B	C	D	CD	CD+B	X
1	0	1	1	1	1	1
1	1	0	0	0	1	1

Karnaugh Maps use SOP → Series of elements added together

Alternative to a Truth table.

$= AB + ACD$ ← SOP

$AB + A(B+C) + B(B+C) = \cancel{AB} + AB + AC + B + BC$



$$\underline{A}\,\underline{B} + AC + B + B\underline{C} = B + AC$$

AB \ C	0	1
00	0	0
01	1	1
11	1	1
10	0	1

1 Value change important

B + AC

$AB = 1$

$AC = 1$

$B = 1$

$BC = 1$

- look for patterns.

Wk 10

CAR – Bool Algebra SOP Form, Karnaugh Maps.

* ~~[illegible]~~

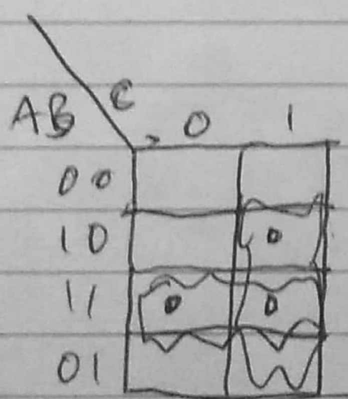
$$X = A \cdot B + A \cdot B' \cdot C + A \cdot B \cdot C$$

$$= A(B + B'C + BC)$$

$$= A(B(1+C) + B'C)$$

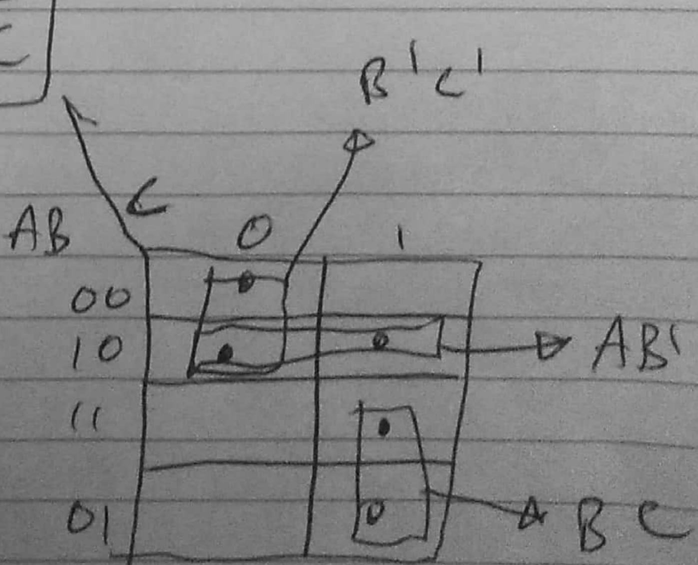
$$= A(B + B'C) = A(B+C) = AB + AC$$

$$AB + AB'C + ABC$$



$$AC + AB.$$

$$= \cancel{\boxed{AB' + B'C' + BC}}$$

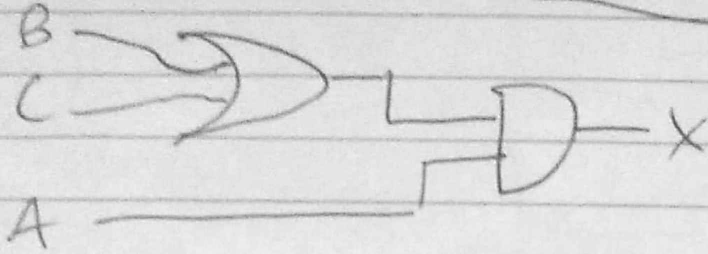


* Distributive Law *

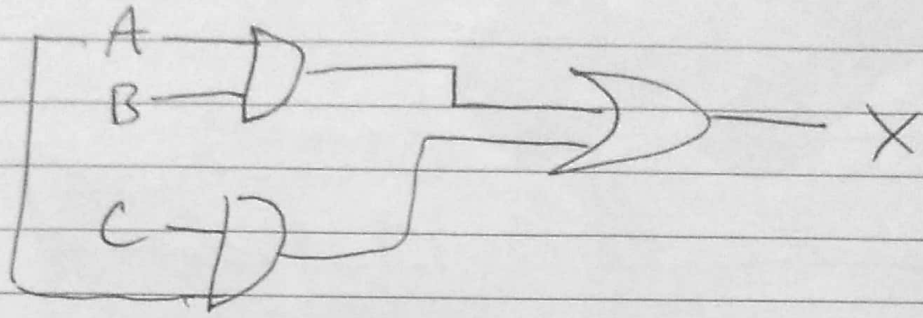
easier to work with

$$A(B+C) = \boxed{AB + AC}$$

SOP and POS Forms

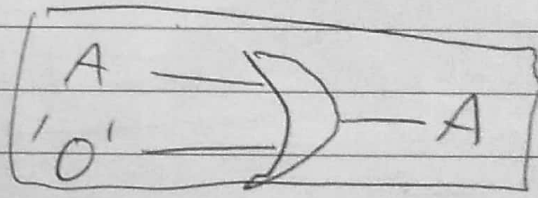


$\equiv$

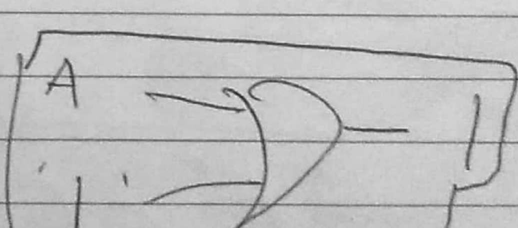


Rules:

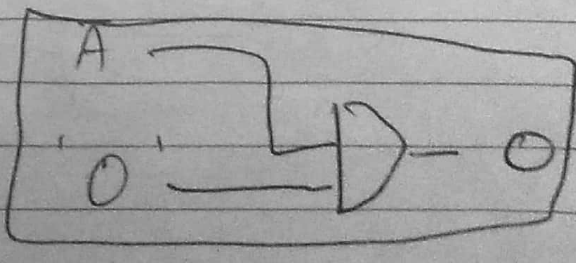
1 – $\boxed{A + 0 = A}$



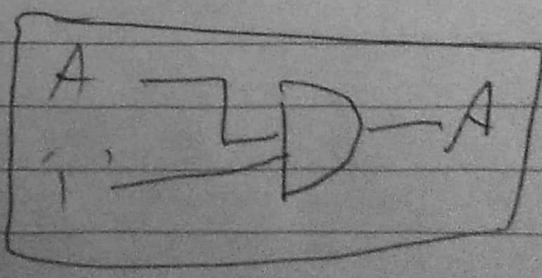
2 – $\boxed{A + 1 = 1}$



3 – $\boxed{A \cdot 0 = 0}$

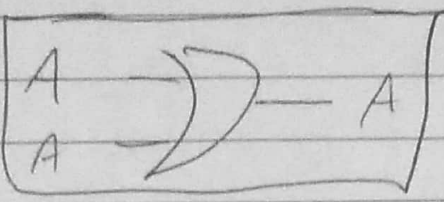


4 – $\boxed{A \cdot 1 = A}$



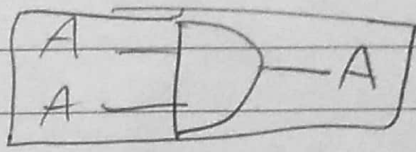
* IDEMPOTENT RULES *

5- $\boxed{A + A = A}$



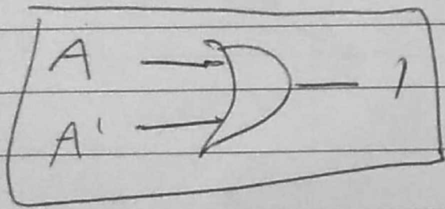
$A + BA + AC + BC$

6- $\boxed{A . A = A}$

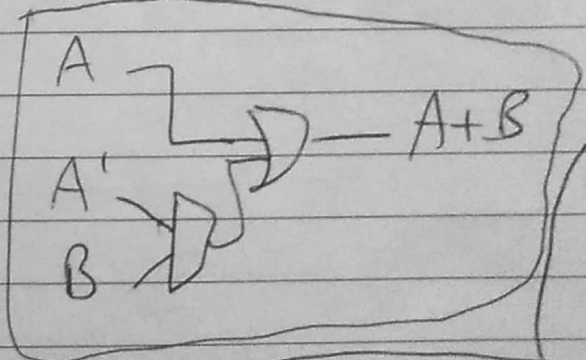


7 - * Laws of Complementarity *

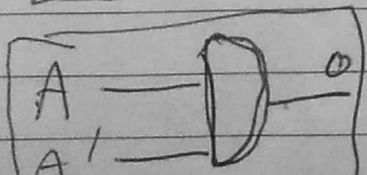
$\boxed{A + A' = 1}$



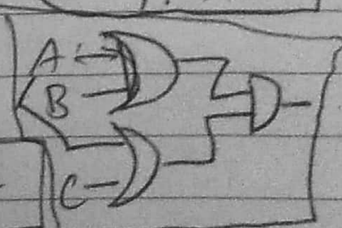
11 - $\boxed{A + A'B = A + B}$



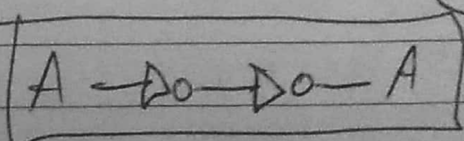
8 - $\boxed{A . A' = 0}$



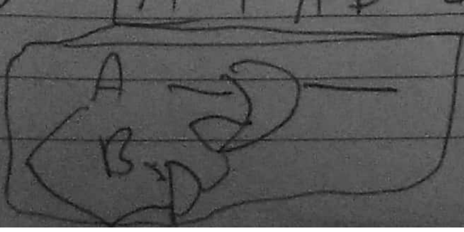
12 - $\boxed{(A+B)(A+C) = A + BC}$



9 - $\boxed{(A')' = A}$



10 - $\boxed{A + AB = A}$ $\to$ $\boxed{A(1+B) = A}$



A	B	AB	X = A + AB
1	0	0	1
0	1	0	0

CAR - Some Maths on Our Circuit

Outside Alarm

$$X = (B \cdot (A+C)) + (C \cdot D') + (E \cdot F')$$

$\Updownarrow$ The same -

Garden Security Light

$$X_1 = (B \cdot (A+C)) + (C \cdot D') + (E \cdot F')$$
$$= (B(A+C)) + (CD') + EF'$$

Warning Message to tenant Manager

$$X_2 = (C \cdot D') + (E \cdot F') = CD' + EF'$$

Power Notification

$$X_3 = (E \cdot F') = EF'$$

$X_1 = (B(A+C)) + (CD') + EF'$

$Y = CD' + EF'$

$Z = EF'$

$X = BA + BC + CD' + EF'$

Karnaugh Maps

Z:

$EF' = 1$

Optimised = EF'

E \ F	0	1
0		
1	1	

Y:

$CD' = 1$

$EF' = 1$

Optimised = $CD' + EF'$

CD \ EF	00	10	11	01
00				
10	1	1	+	+
11	1			
01	X			

CD \ EF	00	10	11	01
00		1		
10	1	1	1	1
11		1		
01		1		

$\underline{X} = BA + BC + CD' + EF'$

ABC \ DEF	000	001	011	111	101	100	110	010
000							1	1
001	1	1	1				1	1
011	1	1	1	1	1	1	1	1
111	1	1	1	1	1	1	1	1
101	1	1	1				1	1
100							1	1
110	1	1	1	1	1	1	1	1
010							1	1

$BA = 1$

$BC = 1$

$CD' = 1$

$EF' = 1$

BC

AB

CD'

EF'

Signed / Unsigned Numbers

Signed Magnitude

↳ Left most, significant bit as the sign

$\boxed{\begin{matrix} 0 = + \\ 1 = - \end{matrix}}$

↳ Remaining bits indicate magnitude of the number in binary

00011 011 = +27
10011 011 = -27

1'S complement

↳ Find the positive signed magnitude notation of a number in binary and invert the bits to gain its negative form in 1's complement

27 = 00011011

↓

-27 = 11100100

2'S Complement

27 = 00011011

$\downarrow$ 1's

11100100

\+ 1

11100101 = -27

Converting Signed binary to decimal

Signed magnitude

- If MSB is 0, Remaining bits can be interpreted and converted as with unsigned binary.
- if MSB is 1, work out the magnitude of the remaining bits.

$1(1001010) = -(1\times64 + 1\times8 + 1\times2) = -74$

1's complement

1 - Invert bits
2 - convert from binary to decimal
3 - Put a minus in front.

11001010
$\downarrow$ 1
$00110101 \xrightarrow{2} 32+16+4+1 = 53 \xrightarrow{3} -53$

2's Complement.

11001010

$-128 + 64 + 8 + 2 = -54$

1, 4, 8

$\frac{11}{2} = 5 \text{ r } 1$

$\frac{5}{2} = 2 \text{ r } 1$

$\frac{2}{2} = 1 \text{ r } 0$

$\frac{1}{2} = 0 \text{ r } 1$

$\frac{29}{2} = 14 \text{ r } 1$

$\frac{14}{2} = 7 \text{ r } 0$

$\frac{7}{2} = 3 \text{ r } 1$

$\frac{3}{2} = 1 \text{ r } 1$

$\frac{1}{2} = 0 \text{ r } 1$

$\frac{114}{2} = 57 \text{ r } 0$

$\frac{57}{2} = 28 \text{ r } 1$

$\frac{28}{2} = 14 \text{ r } 0$

$\frac{14}{2} = 7 \text{ r } 0$

$\frac{7}{2} = 3 \text{ r } 1$

$\frac{3}{2} = 1 \text{ r } 1$

$\frac{1}{2} = 0 \text{ r } 1$

$\frac{111}{2} = 55 \text{ r } 1$

$\frac{55}{2} = 27 \text{ r } 1$

$\frac{27}{2} = 13 \text{ r } 1$

$\frac{13}{2} = 7 \text{ r } 1$

$\frac{7}{2} = 3 \text{ r } 1$

$\frac{3}{2} = 1 \text{ r } 1$

b $\frac{1}{2} = 0 \text{ r } 1$

Scanned by CamScanner

Wk 13

Computer Structure and Function. CAR

CW - Poster Design on "How do computers do what they do?"

Computer Systems
- → Architecture → Design, logical execution.
- → Organization. → 'Building a house by bricks and technology'
  - → More lateral
  - → physical makeup.

Computer Functions.

- Data processing
- Data storage → Short term, → long term
- Data movement → I/O, → Peripheral, → Data Communication
- Control

Operating environment

Data movement apparatus

Control mechanism

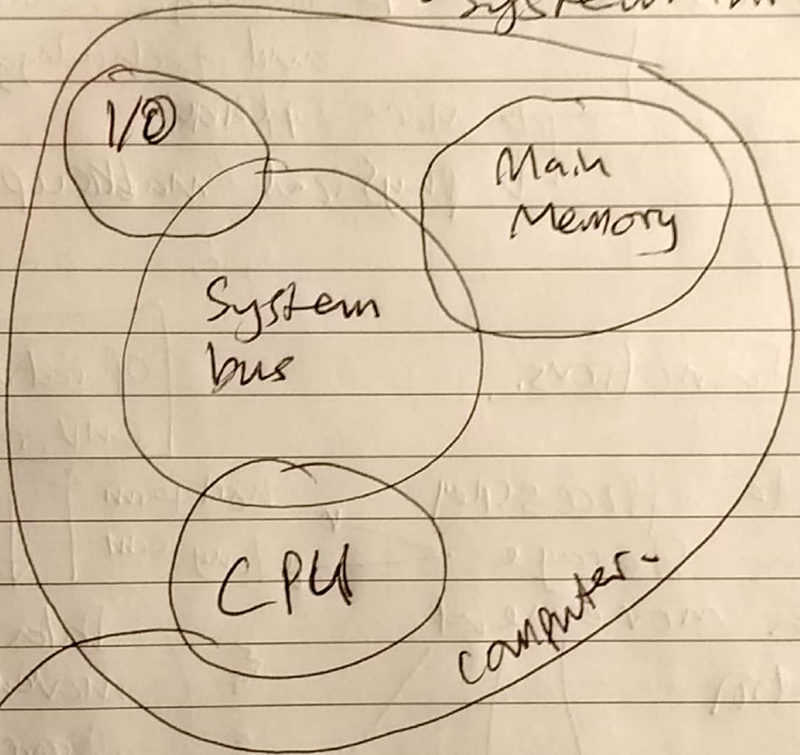
Data Storage facility

Data processing facility

Computer Structure

Interrelated components

Integral components → CPU
→ Memory
→ I/O devices.
→ System interconnection.
(buses)



Von Neumann Machine
↳ Stored program
↳ IAS computer.

CPU

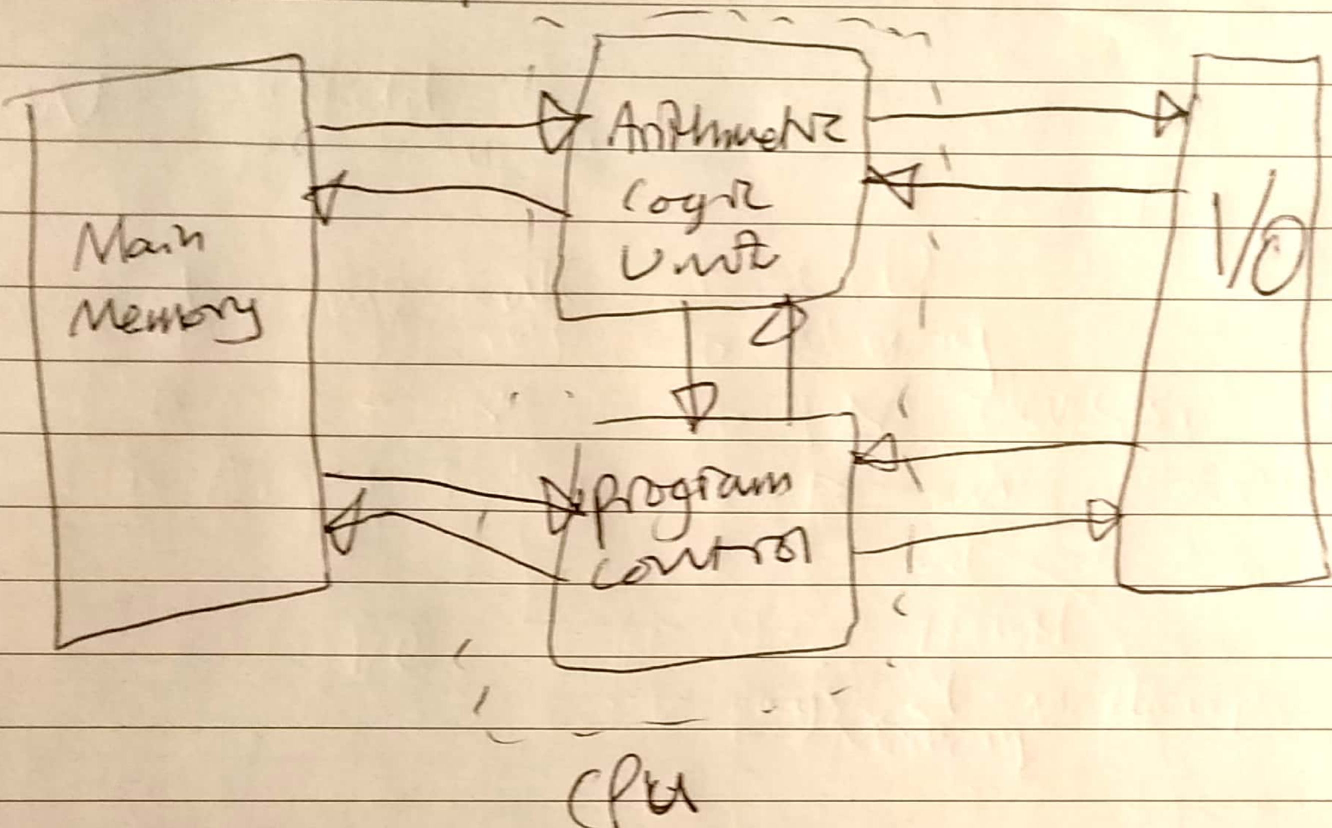
- Control Unit
- Arithmetic / Logic Unit ALU
- Registers → temporary stores for incoming data.
- CPU Interconnections

Sequencing logic

Control unit registers and decoders

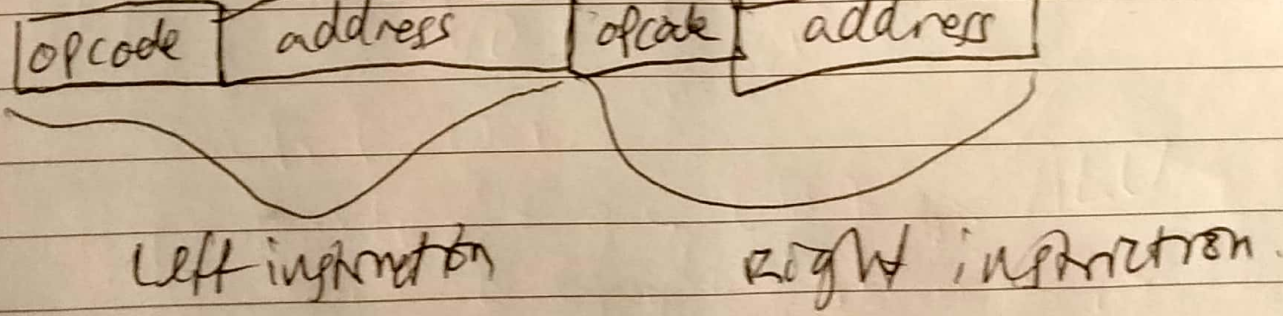
Control memory

IAS Computer Structure



Memory
↳ 1000 locations
↳ 40 bit words.

Number words and instruction words.



WK14 CAR - IAS Computer

CA - logical structure
CO = techology behind.

Von Neuman - basic architecture.

Computer structure - CPU
- Memory
- I/O
- Busses

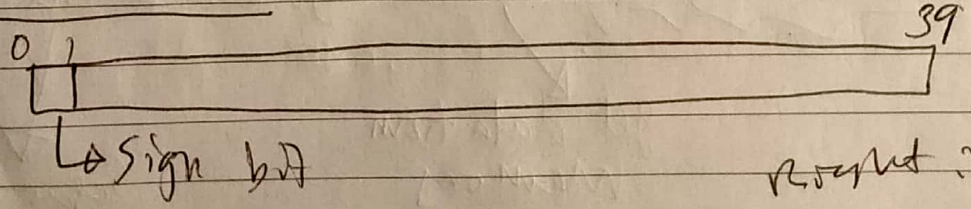
CPU - control unit
- ALU
- Registers
- internal bus.

IAS structure

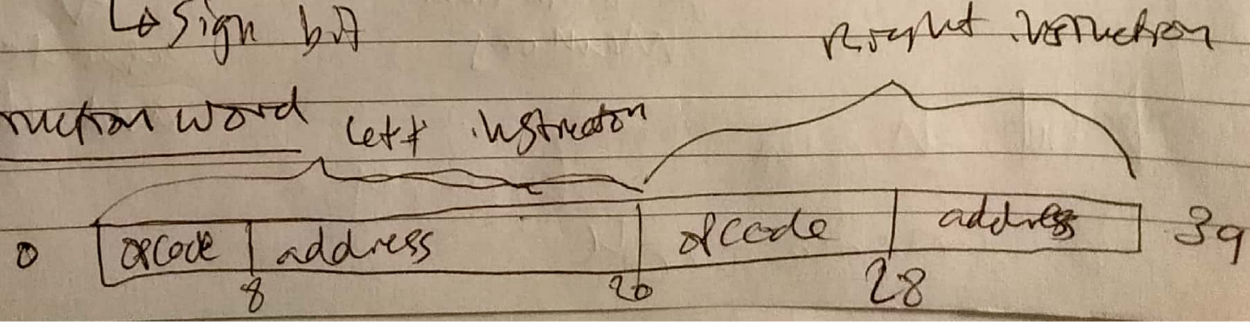
↳ stored-program concept

Memory structure : - 1000 locations
- Words (40) bits - binary

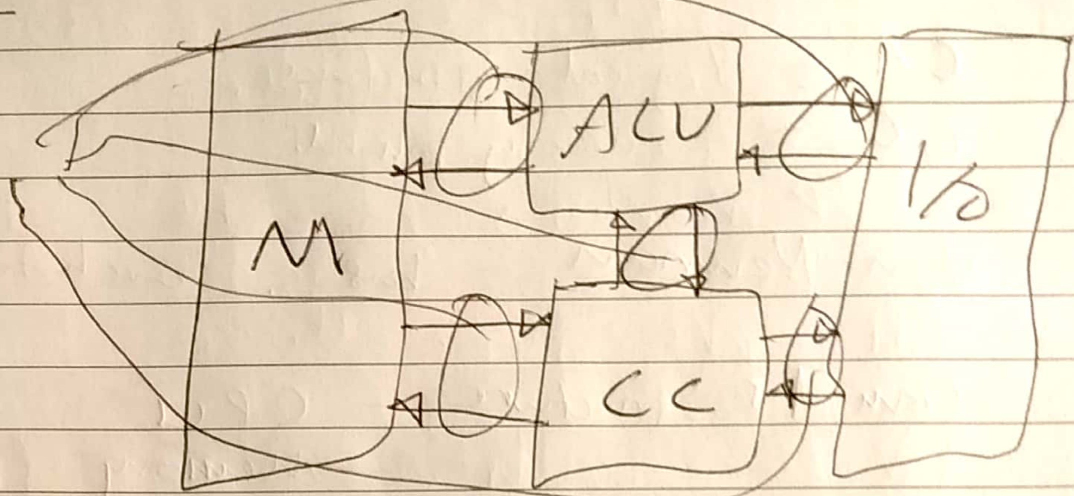
number words



instruction word



Registers



- Small temporary memory.

egs

- Memory Buffer register (MBR)
- memory address Register (MAR)
- Instruction register (IR)
- Instruction Buffer Register (IBR)
- Program counter (PC)
- Accumulator (AC) & multiplier Quotient (MQ)

* Look up ALU / CC diagram. *

Important aspect with von Neumann
- execution in sequence.

Instruction cycle: fetch cycle, Execute cycle.

fetch cycle → get data from memory

Execute cycle → Separate opcode and address.

→ Put it in

Da course work

Instruction Sets.

↳ The relation between a function and opcode.

Fundamental Operations

- Data transfer
- unconditional Branch
- conditional branch
- Arithmetic
- Modify address

IAS key concepts

↳ single read - write memory
  ↳ data and instructions

↳ Addressable Memory

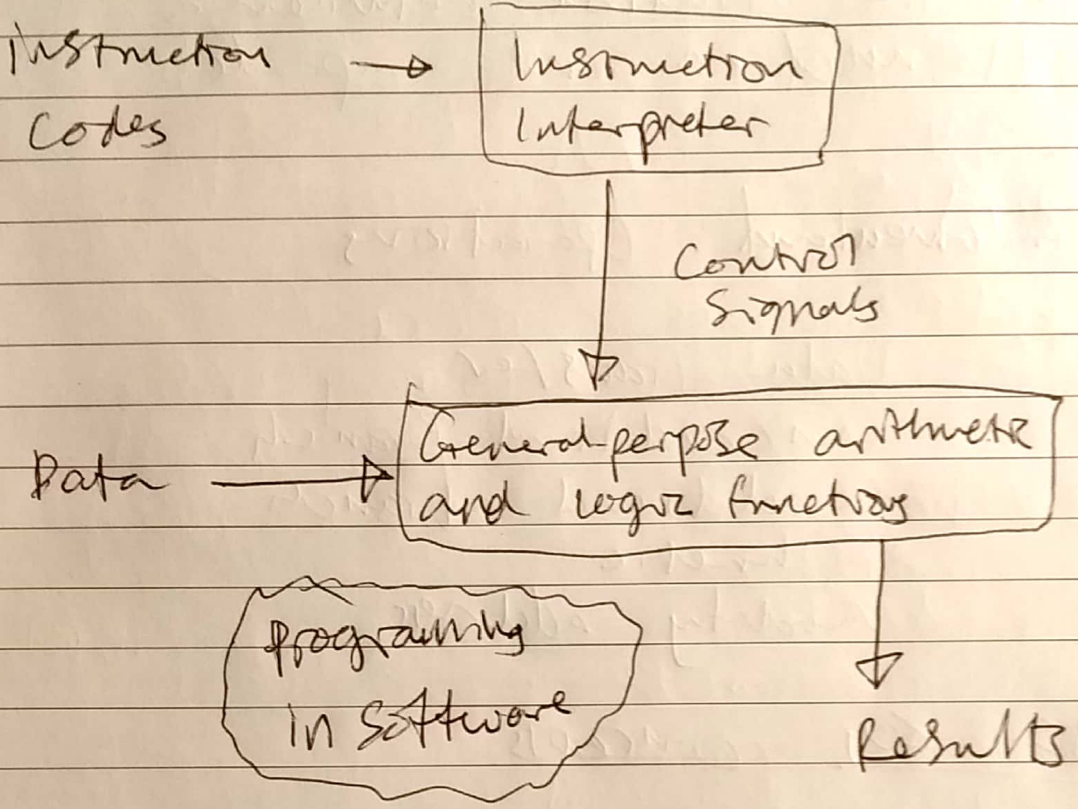
↳ sequential execution (unless explicit modification) from one instruction set to the next.

Hardwired program

Data → [Sequence of arithmetic and logic functions] → Result

Programming in hardware

Hardwired
~~HW~~ Program Vs SW



be fully familiarised with registers by next week.

CW - Easy, Poster design.

- General purpose Circuit
- Fetch - Execute Algorithm -
- 9th February.

CAR Wk15 Fetch-Execute Cycle

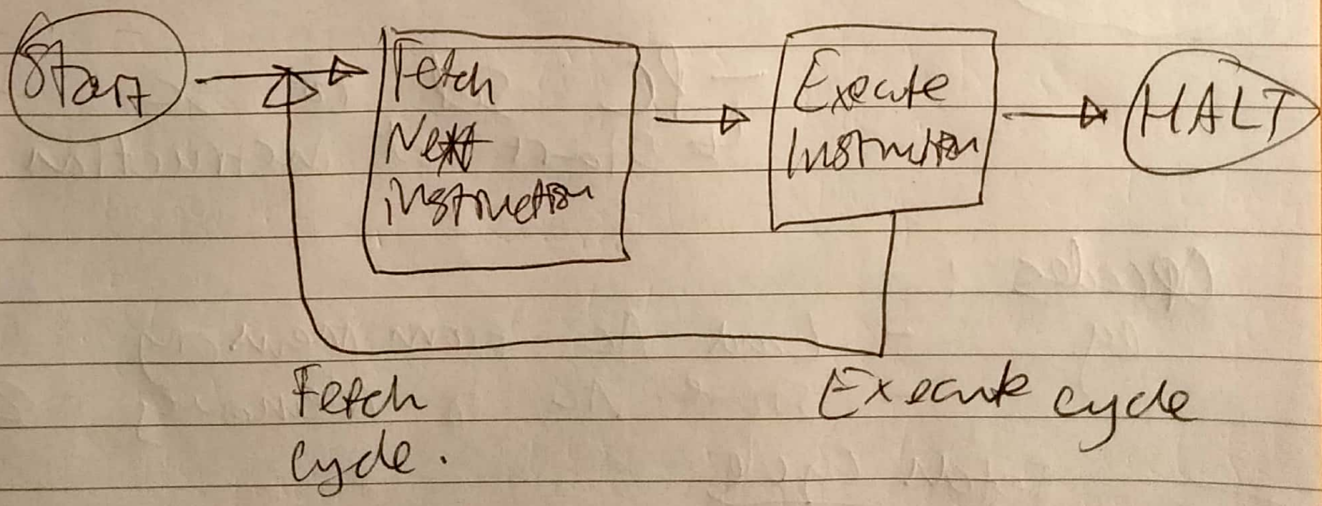
Registers: MBR, MAR, IR, IBR, PC, AC, MQ

Hardwired programs Vs Software

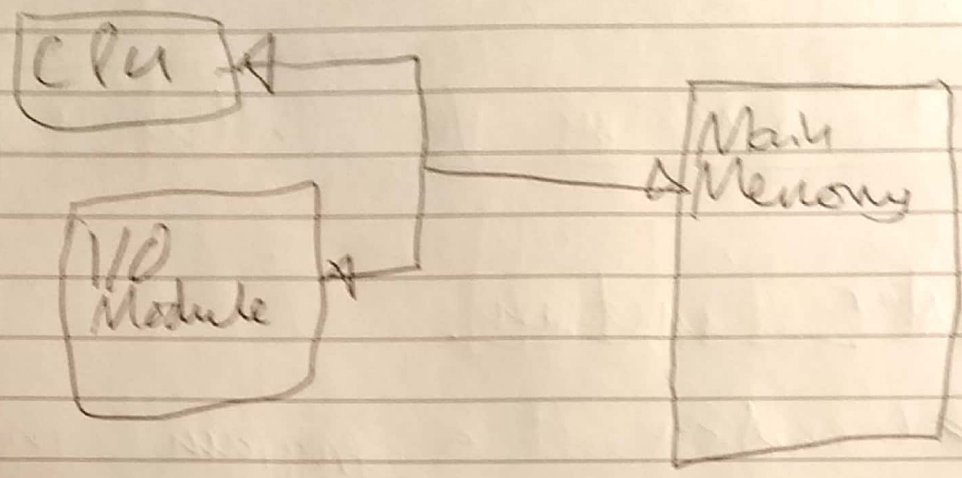
Computer function

- Execution of a program, instruction set
- Processor → Fetch, read instructions
  - → Execute, perform instructions
- Program execution
  - → repeating F/E cycle.

NF×
CAF✓
INTR✓
WEBF✓
IND✓



Instruction fetch + Execute



Add Marking comments for your lit review

- Think about the number of op codes and how many words of memory can be addressed

- PC set to x
  ↳ start at instruction x

Opcodes:
eg
- Load AC from memory = 1 (0001)
- Store AC to memory = 2 (0010)

Fetch cycle

Step 0 - Load from memory.
- PC says go to memory location.

Step 1 - MAR reads from main memory
Bring to MBR

Execute cycle

Step 2 — ~~Send~~ Go to memory and load address in the IR to the AC from the MBR.

Step 3 Fetch

- Interpret
- PC say another instruction coming

Step 4 Execute

- Conduct operation.

Step 5 Fetch Memory → CPU Register.

Step 6 Execute - Storage to memory.

~~Three cycles.~~

*** Lets be honest, a lot of this ~~[illegible]~~ went over your head. Make an effort to go over it ~~and~~ understand the process, including the involvement of registers, in detail. ***

CW3 Notes - "Designing an Adder/ Subtractor Machine."

$\boxed{A2}$

- Based on IAS computer.
- Include Memory, cache, ALU Registers and Buses

8-bit memory

- include fetch execute algorithm. → 3 F/E cycle egs.

WK16

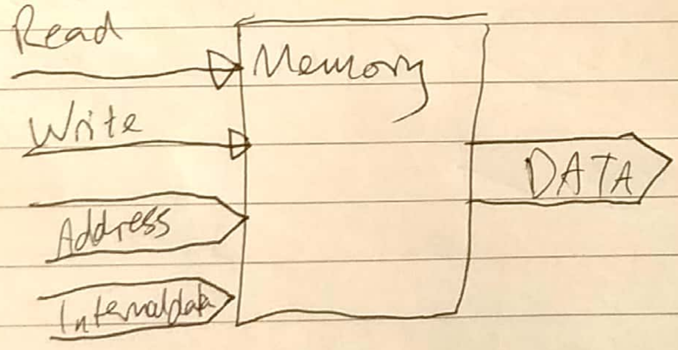
CAR Interconnections + Memory

Interconnection Structure

- Computer - network of basic modules
- Paths for connecting modules
- Structure dependant on exchanges.
- Quality of which affects performance.

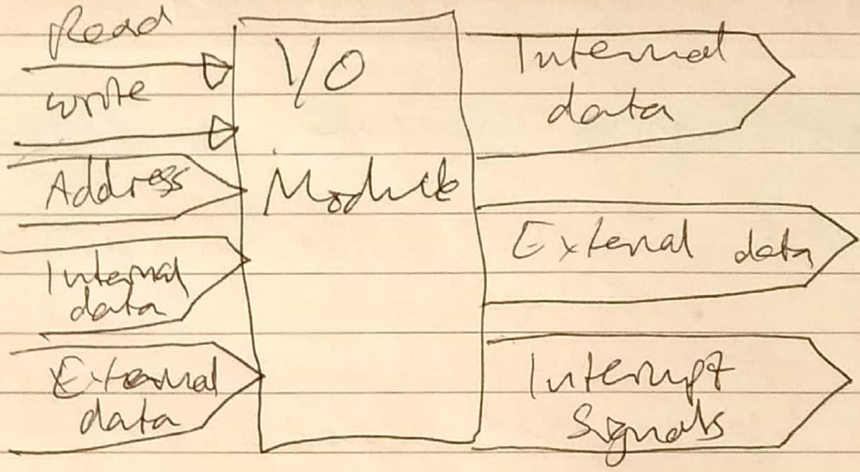
Memory

- N words of equal length
- Unique numerical address
- Read/Write data word.
- Operation indicated by read and write control signals.



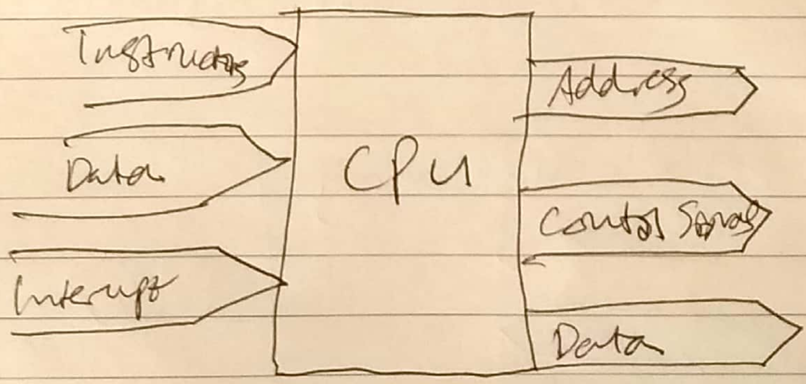
I/O Module

- Similar structure to Memory
- Control more than one extend device, port - unique address
- External data paths, extend device
- Interrupt signals → stop the main stream of processing.



Processor

- ↳ reading instructions and data
- ↳ write out data after processing
- ↳ Use control signals
- ↳ receive interrupt signals



Interconnection Structure

must support transfers

- ↳ memory to processor
- ↳ processor to memory
- ↳
- ↳

- Bus interconnection
- Point to point interconnection

Bus Interconnection

↳ Comm pathway connecting two or more devices.

↳ Shared transmission medium.

↳ Signal overlap and become garble

↳ only one device at a time.

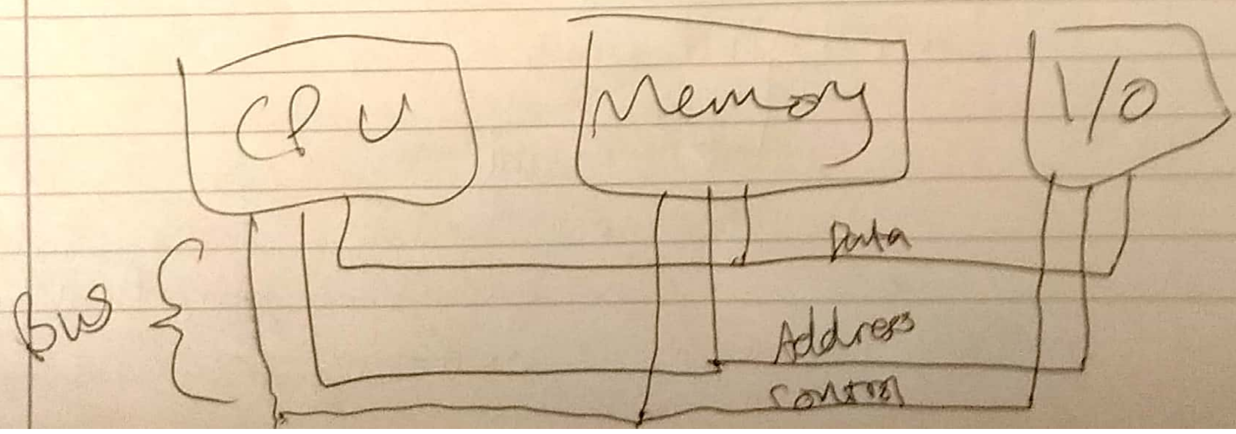
↳ Multiple communication lines

↳ each transmission signals representing binary 0/1

↳ Data lines

↳ Address lines

↳ Control lines



Data lines

- More data
- data bus
- Bus width - Number of Lines
  - key in determining System performance

Address Lines

- Source / destination of data on data bus
  - Address bus
- Understands difference between data and locations
- Bus width determines maximum possible memory capacity.
- Address I/O Ports.
- Higher order bits → Select Module
  - Lower order bits → Memory location or I/O Ports.

Control Lines

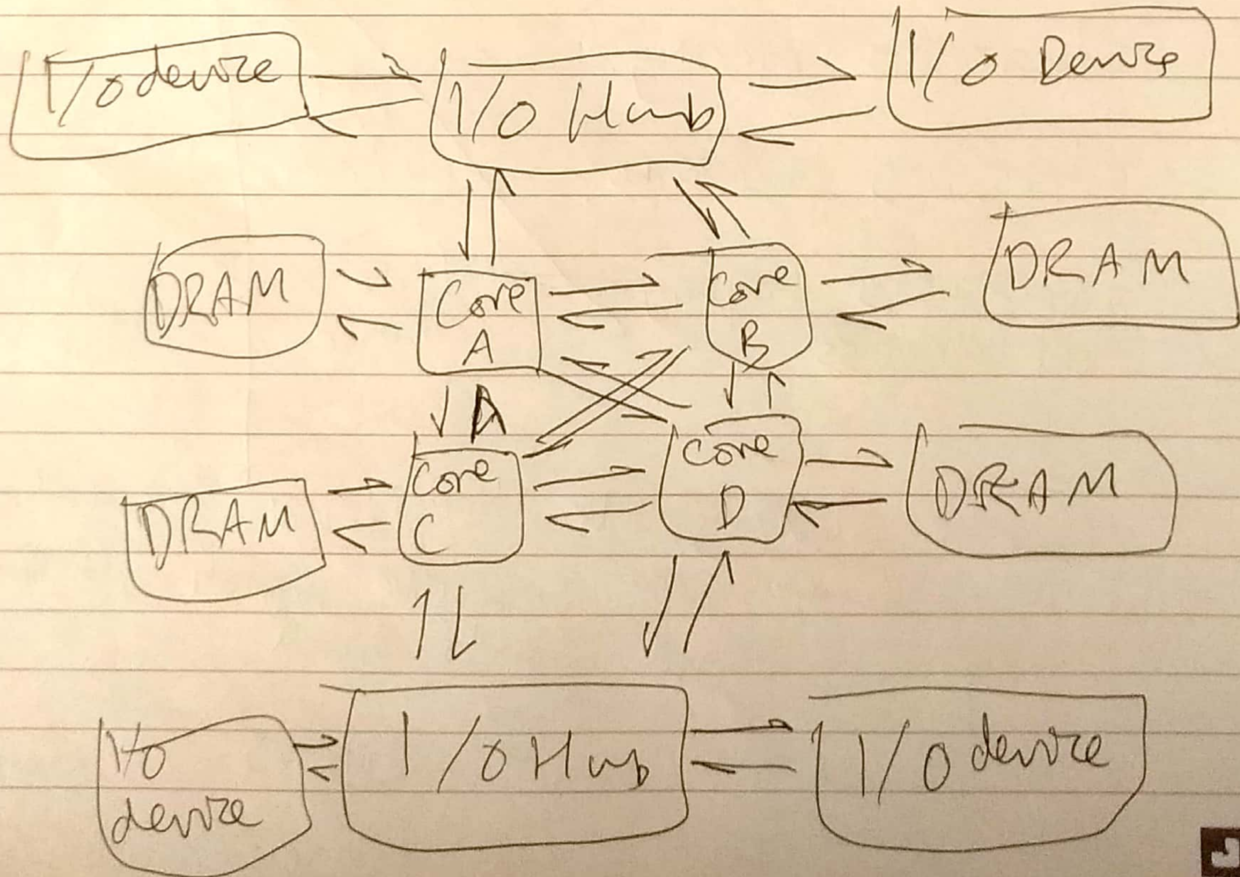
- Controls access and use of data and address bus.
- Control Signals
  - Command Signals, specify operation
  - Timing Signals, validity of data and address information.
- Control lines include: Memory R/W

Bus Operation

- Sending data
  - Obtain use of the bus
  - Transfer data via bus
- Request data
  - Obtain use of the bus
  - Transfer a request to other module over appropriate control/address lines.

Point - to - Point interconnection

- contemporary
- Lower latency, higher data rates, better scalability
- QPI - 2008

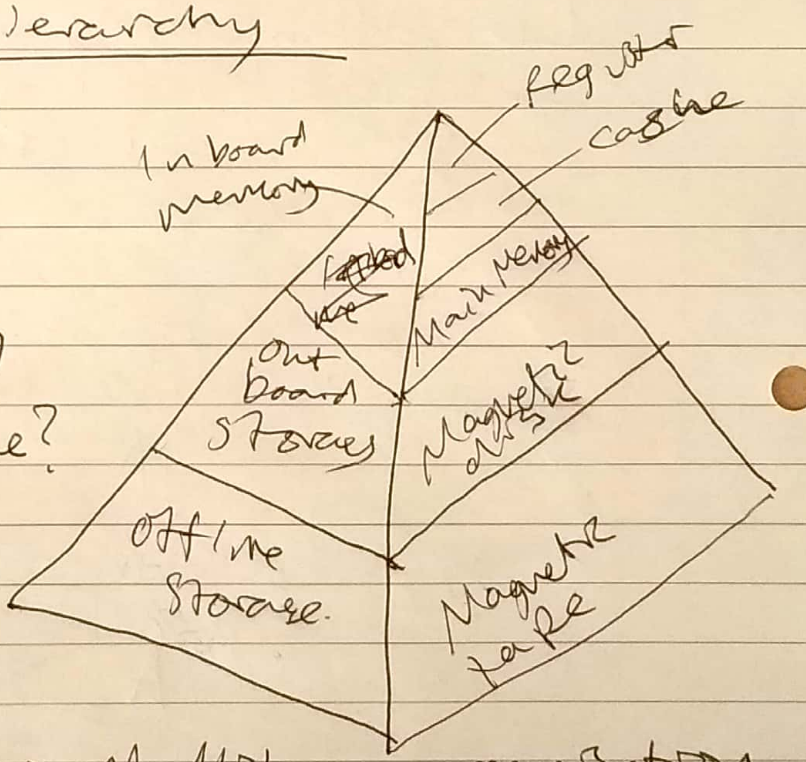


Memory
↳ store binary values
↳ key
↳ capacity
↳ Access time
↳ cost

Greater capacity → Smaller cost per bit
Greater capacity → slower Access time
Faster Access time → Greater cost.

Memory Hierarchy

- Decreasing cost per bit
- Increasing capacity
- Increasing access time?
- Decreasing frequency of access



Cache Memory → Multilevel organisation.
↳ Fast/Expensive memory with less expensive and lower speed.
↳ Contains copy of portions of memory.
↳ Faster access time.

WK 19

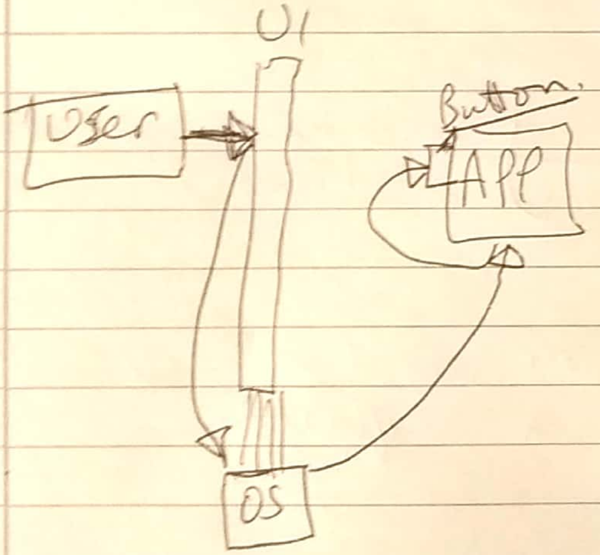
Exam
↳ CPU Scheduling

CAR: OS overview / I/O

CW → Questions to consider

- ↳ Pull it apart before learning it.
- ↳ Process management
- ↳ Memory management
- ↳ File management.
- ↳ I/O Management.

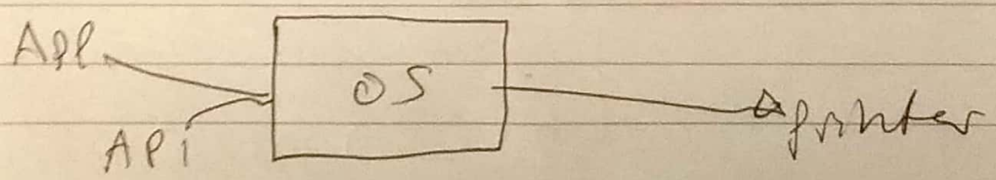
- Step out of the detail.

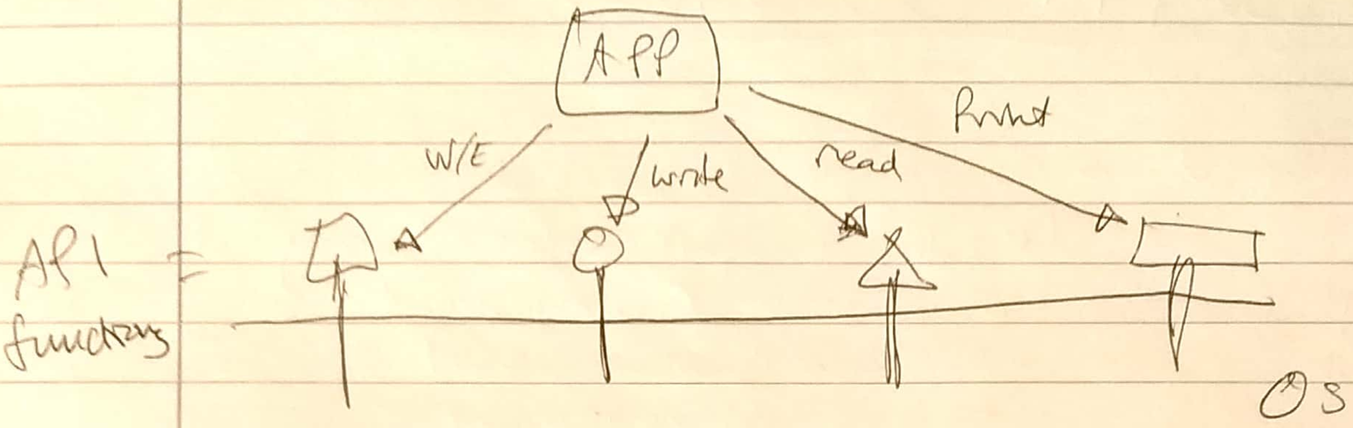


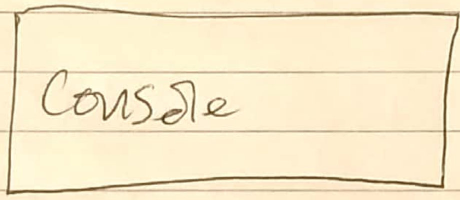
OTHER SYSTEMS.

Application Programming Interface API

- App sits as an Island on top of the operating system







Process Scheduling → how you organise that.

Process control Block

↳ Gathers information about a task.
↳ dumped into a queue.

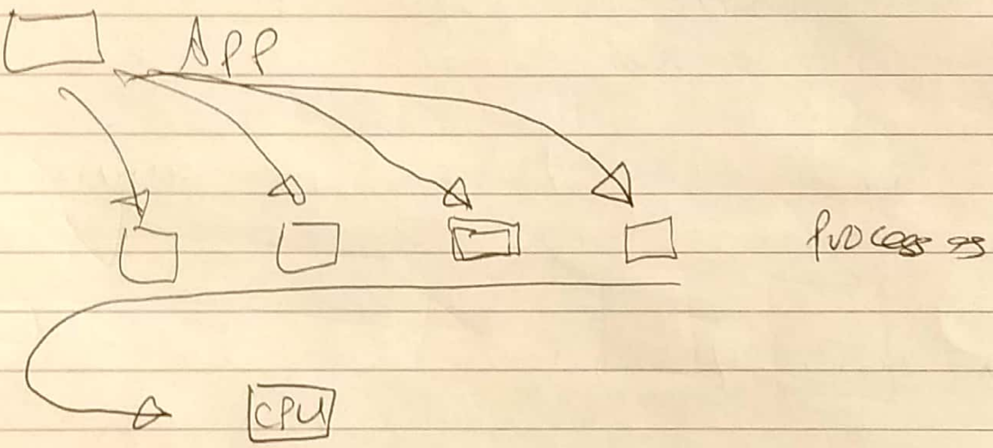
- Multitasking !
  ↳ Algorithms.
    ↳ Round-robin
      ↳ Fixed amount of time for each process - time quanta -
      ↳ PREEMPTIVE
    ↳ FCFS - First come first serve
      ↳
    ↳ SJF - Smallest Job First.
      ↳
    ↳ PRIORITY
      ↳

wk20

CAR -

CW - Doesn't need to work.
↳ cover concepts - creatively.

Process is a way an application packages data for the CPU



Throughput

Time Quanta.

PREEMPTIVE / NON PREEMPTIVE ALGORITHM

Interrupt / Process driven.

→ Average job length?

Round-Robin — a bunch of things doing stuff in a circle.

Process Manager - defines the time quanta.
Indivisible unit of time
of which is allowed to a
process before a switch.

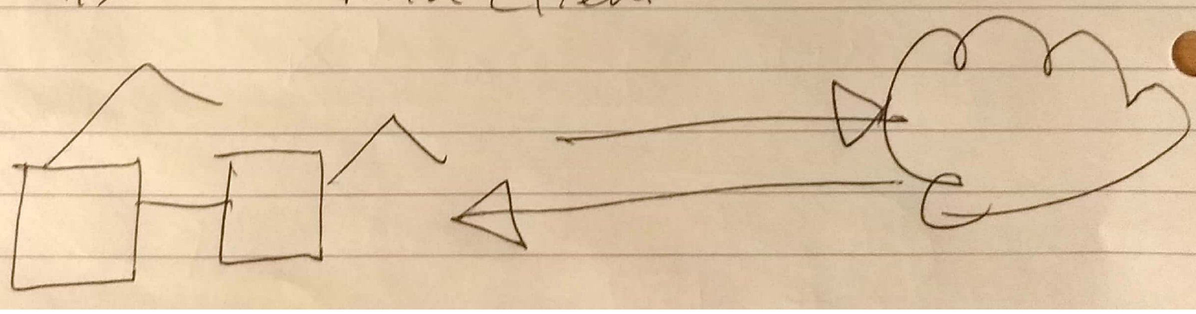
Be imaginative, I guess :-

CW

↳ How are processes managed?

Media player, Word document management, Browser..

Virtual Desktops
Terminal Service
CITRIX - Thin Client



WK21

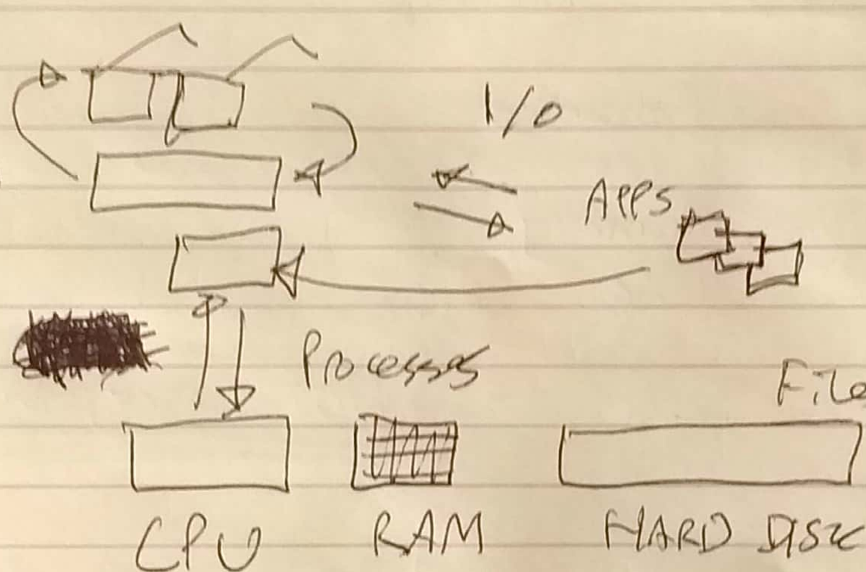
CAR: COURSEWORK

- Paging Algorithms → Relocation of data due to RAM limits.
- Main Memory Management.
- Indicate how these concepts would work.
- How is it distributed?

Page Replacement Algorithms

Distributed Architecture

Managing Live Processes



Idea → Prioritisation of data loss prevention?

- → Local CPU → Cloud Storage.
- → Thin client / virtual desktop - Terminal service
- → Server Farms. — Network Load balancing.

* Talk sensibly about it.
  - → How well do you justify it?

ID problem → Formulate Solution → Justify Solution

→ Vocabulary forward.

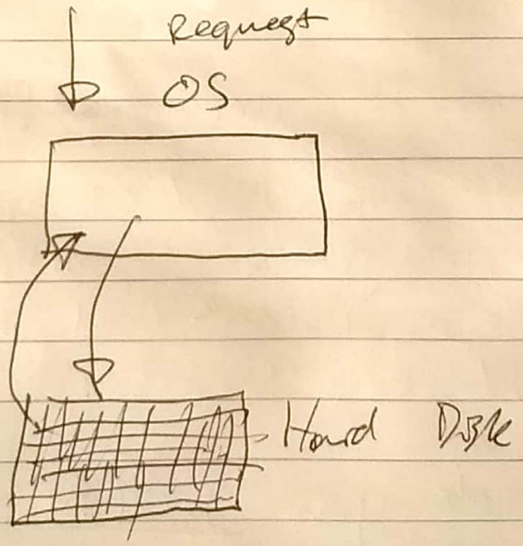
Presentation

Scenario based → hypothetical.
- on a train
- at work
- MMO
- Walking around a city

4 Questions
- 1 - Processes
- 2 - Files - open/close/save (write) - how?
- 3 - Memory
- 4 - I/O + App.

Files

FOPEN (" ")



Framework / Guidance

Marking Criteria

Questions

NOT - Descriptive - CW

Be:
- Analytic
- Synthetic

→ Justification

New concepts
→ Response to a problem.