⊛ Digital Integrated circuits and CMOS Technology

- Implementation of algorithm:

The implementation of any algorithm will fail into two main categories

① Hardwired algorithms: A special purpose circuit is designed to compute the required algorithm. This is done as a special purpose silicon chip called an ASIC (Application Specific Integrated circuit)

- The algorithm is embedded within the circuit hardware, The hardware is completely specific to one particular algorithm

② Computation in Software: This uses a general-purpose piece of hardware (e.g µP).

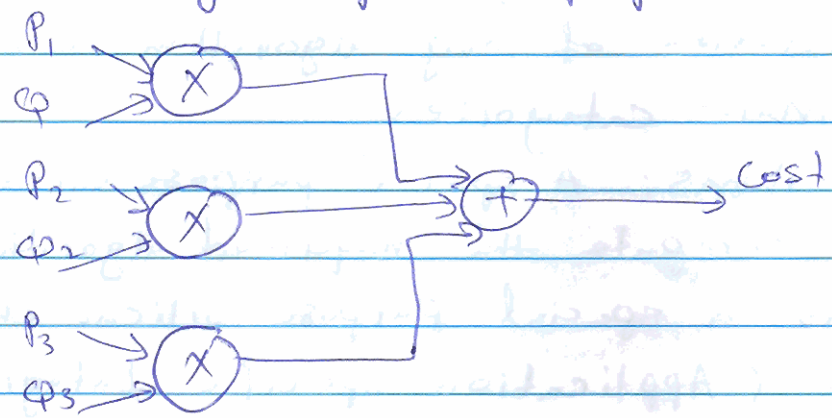
- This uses the capability of computing almost any algorithm
- The algorithm is expressed as a series of instructions executed by hardware.

Ex: prices of shopping list

	Price	Quantity
Item 1	P_1	Q_1
Item 2	P_2	Q_2
Item 3	P_3	Q_3

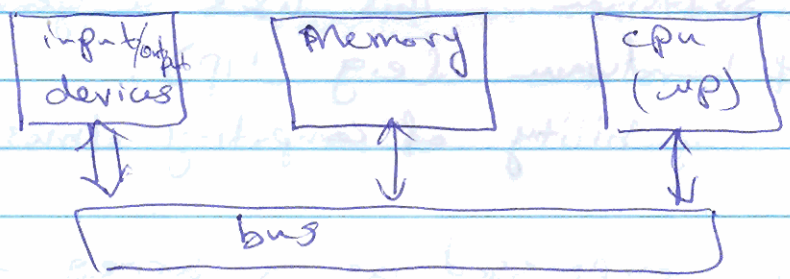
$\Rightarrow$ cost $= P_1 Q_1 + P_2 Q_2 + P_3 Q_3$

① Building a special purpose circuit



* we use 3 multipliers, 1 adder
The answer is produced within 2 steps.

② using a general purpose computer (µP)



- we use these elements to compute any algorithm
- the answer will be produced in 4 steps

step1 : multiply $P_1 * Q_1$
step2 : " $P_2 * Q_2$
step3 : " $P_3 * Q_3$
step4 : Add results together

⊛ Evaluation

- The hardwired algorithm is faster
  (imagine that we want to compute the price of 100 items).

- modification: In ASIC we have to design a completely new chip.
  In software we only need to write another program.
  ⟹ computers (μP) are very flexible
  ASICs are not.

In general:
when we need an extreme speed and ~~lack of~~ inflexibility and lack of modifiability ~~is~~ be tolerated ⟹ use ASICs
Otherwise ⟹ use general μP. (software)

⊛ Using Programmable Hardware

⊛ There are some forms of hardware that are modifiable such as FPGA and PLD.
- These provide greater flexibility than ASICs (but more expensive & less speed)
- These provide greater speed than software

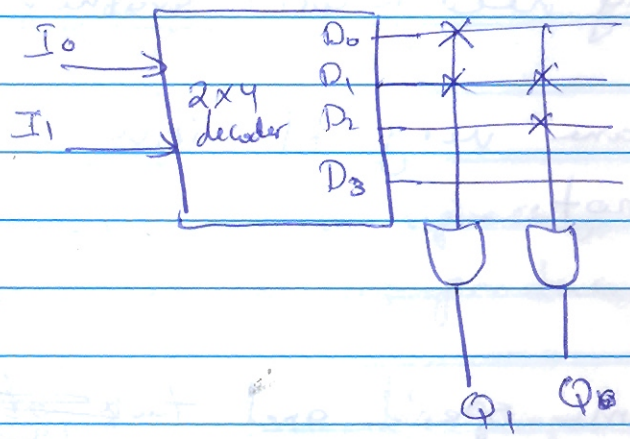
PLD (Programmable Logic Devices)

mainly there are 3 types

(1) PROM (Programmable Read Only Memory)

Fixed AND array, Programmable OR array

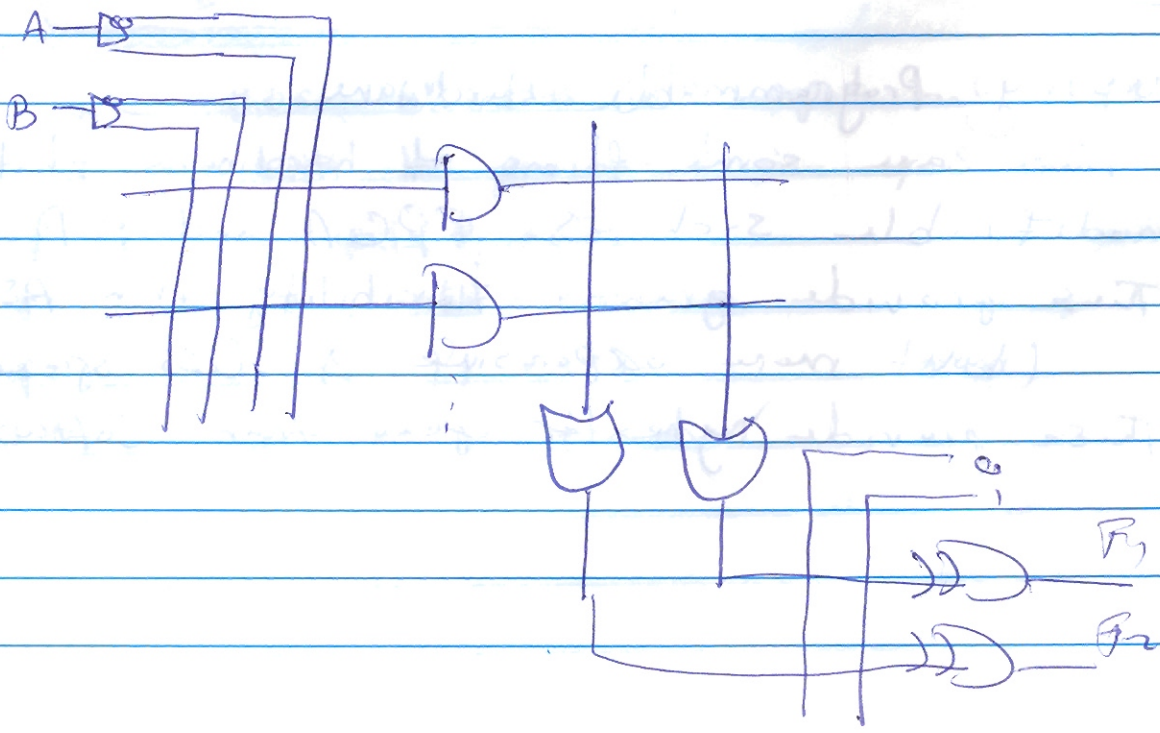
e.g 4 x 2 PROM



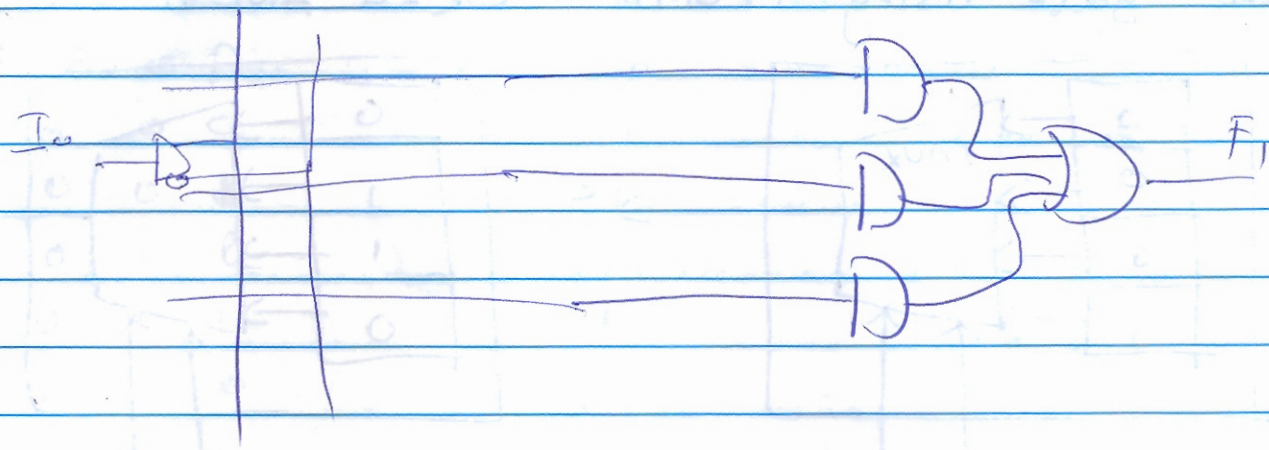
I_0	I_1	O_1	O_0
0	0	1	0
0	1	1	1
1	0	0	1
1	1	0	0

(2) PLA (Programmable logic array)

Programmable and array, Programmable or array



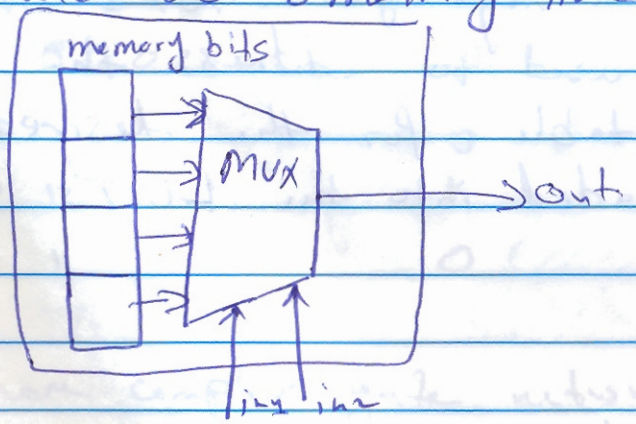
(3) PAL : (Programmable Array logic)



(4) FPGAs (Field Programmable Gate Array)

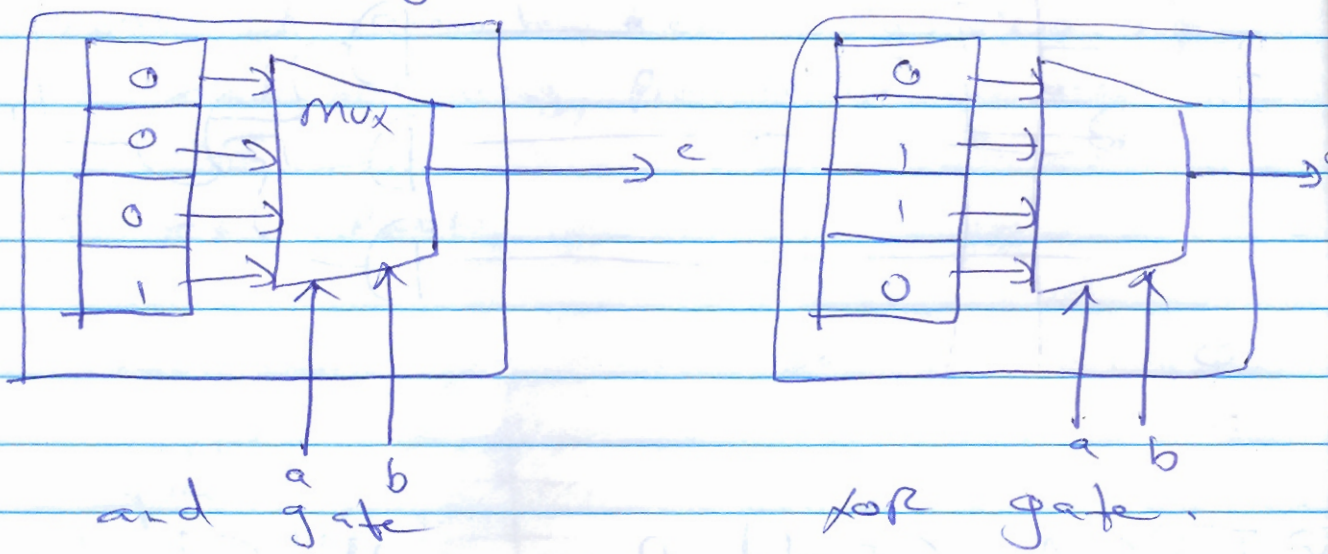
- FPGAs is usually based on a look-up table (LUT) approach.

For example a two input programmable logic gate would look something like this



In this example, the look up table (LUT) is a series of four memory bits. The two inputs (In1 & In2) select one of the memory bits to be fed to the output.

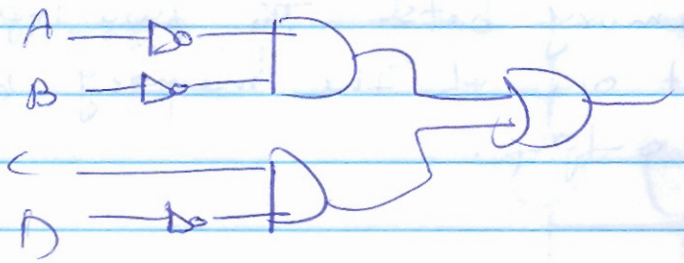
For example if we want to implement an and gate using FPGA c<=a AND b.



⊛ For example in ALTERA FLEX 10K, the gate ~~level~~ logic is implemented using a (LUT). The LUT is a high speed 16×1 SRAM
⇒ Four inputs are used to address the LUT memory. The truth table for the desired gate network is loaded into the LUT's SRA during programming.

Ex:

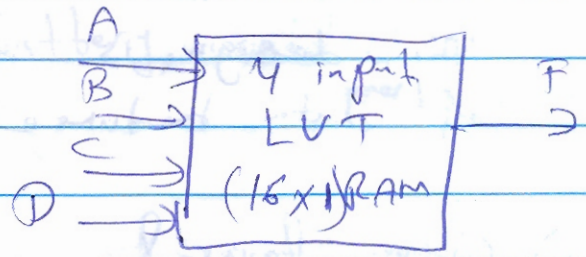
$F = A'B' + CD'$



*

∴ RAM Contents

Address				Data
A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0



⊛ more complex gate networks can be fed to a D flip-flop and then to interconnection network