ROMs, PLAs and FPGAs

October 5, 2006

Why Programmable Logic?

Programmable logic technologies:

- Read-Only Memory (ROM)
- Programmable Logic Array (PLA)
- Programmable Array Logic (PAL)
- Field Programmable Gate Arrays (FPGA)

Why Programmable Logic?

- Facts:
  - It is most economical to produce an IC in large volumes
  - Many designs required only small volumes of ICs
- Need an IC that can be:
  - Produced in large volumes
  - Handle many designs required in small volumes
- A programmable logic part can be:
  - made in large volumes
  - programmed to implement large numbers of different low-volume designs

Some Characteristics

- Permanent - Cannot be erased and reprogrammed
  - Mask programming
  - Fuse
  - Antifuse
- Reprogrammable
  - Volatile - Programming lost if chip power lost – Single-bit storage element
  - Non-Volatile
    - Erasable (EPROM with UV light)
    - Electrically erasable (EEPROM)
    - Flash (as in Flash Memory)

Programmable Logic

- Read Only Memory (ROM) – a fixed array of AND gates and a programmable array of OR gates.
- Programmable Array Logic (PAL) – a programmable array of AND gates feeding a fixed array of OR gates.
- Programmable Logic Array (PLA) – a programmable array of AND gates feeding a programmable array of OR gates.
- Complex Programmable Logic Device (CPLD)/Field- Programmable Gate Array (FPGA) - complex enough to be called "architectures"

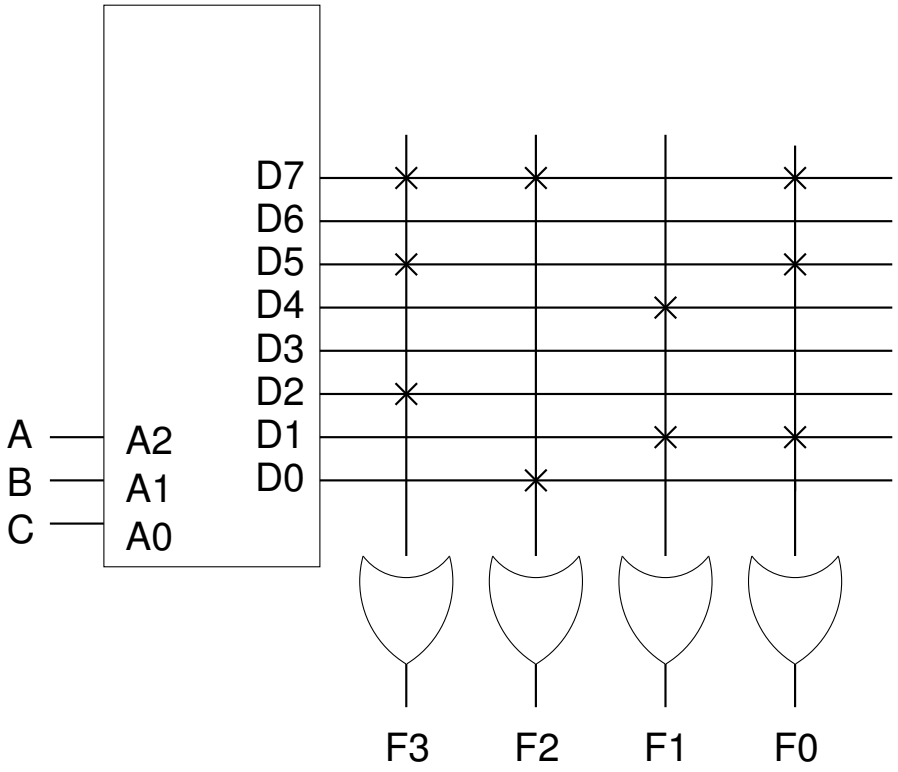
ROMs

- Read Only Memories (ROM) or Programmable Read Only Memories (PROM) have:
  - N input lines, M output lines, and 2^N decoded minterms.
- Fixed AND array with 2^N outputs implementing all N-literal minterms.
- Programmable OR Array with M outputs lines to form up to M sum of minterm expressions.
- A program for a ROM or PROM is simply a multiple-output truth table
  - If a 1 entry, a connection is made to the corresponding minterm for the corresponding output. If a 0, no connection is made.
- Can be viewed as a memory with the inputs as addresses of data (output values), hence ROM or PROM names!

An Example

Example: A 8 $\times$ 4 ROM (N=3 input lines, M=4 output lines)

- The fixed "AND" array is a "decoder" with 3 inputs and 8 outputs implementing minterms.
- The programmable "OR" array uses a single line to represent all inputs to an OR gate. An "X" in the array corresponds to attaching the minterm to the OR
- Read Example: For input (A_2,A_1,A_0) = 001, output is (F_3,F_2,F_1,F_0) = 0011.

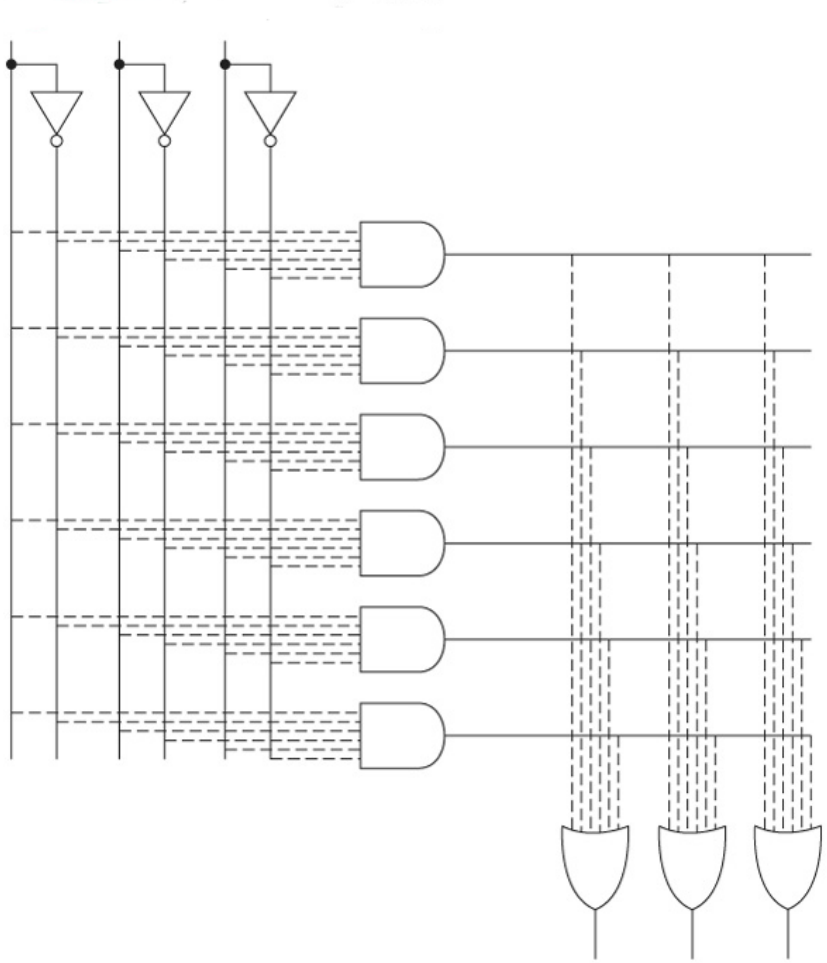


What are functions F_3, F_2 , F_1 and F_0 in terms of (A_2, A_1, A_0)?

PLAs

- Programmable Logic Arrays (PLAs) implement a number of Sum-of-Product expressions
- PLAs are specified by
  - the number of inputs (variables)
  - the number of product terms (AND gates)
  - the number of functions (OR gates)
- Every input variable (or its complement) may be included in a product term by setting a programmable switching element.
- Every function (output) can include **any or all** of the product terms by setting a programmable switching element.

PLA Structure

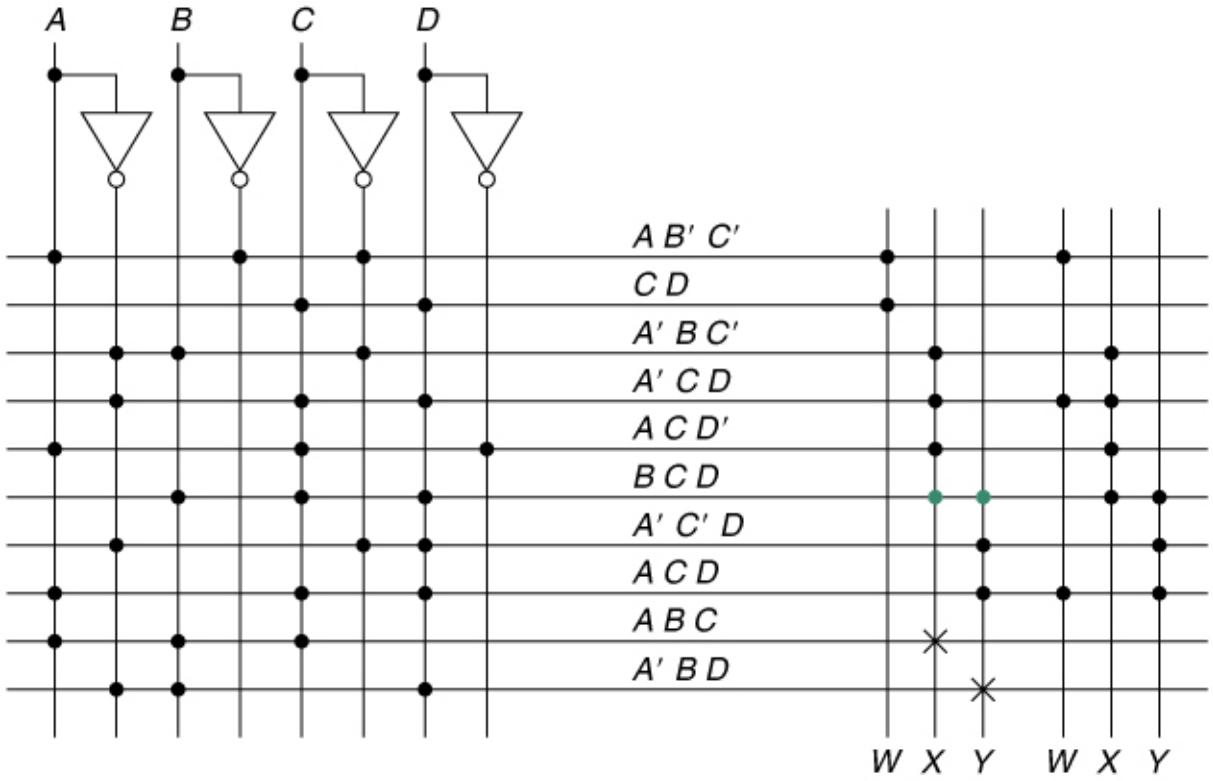


PLA Example

W = AB'C' + A'CD + ACD

X = A'BC' + ACD' + A'CD + BCD

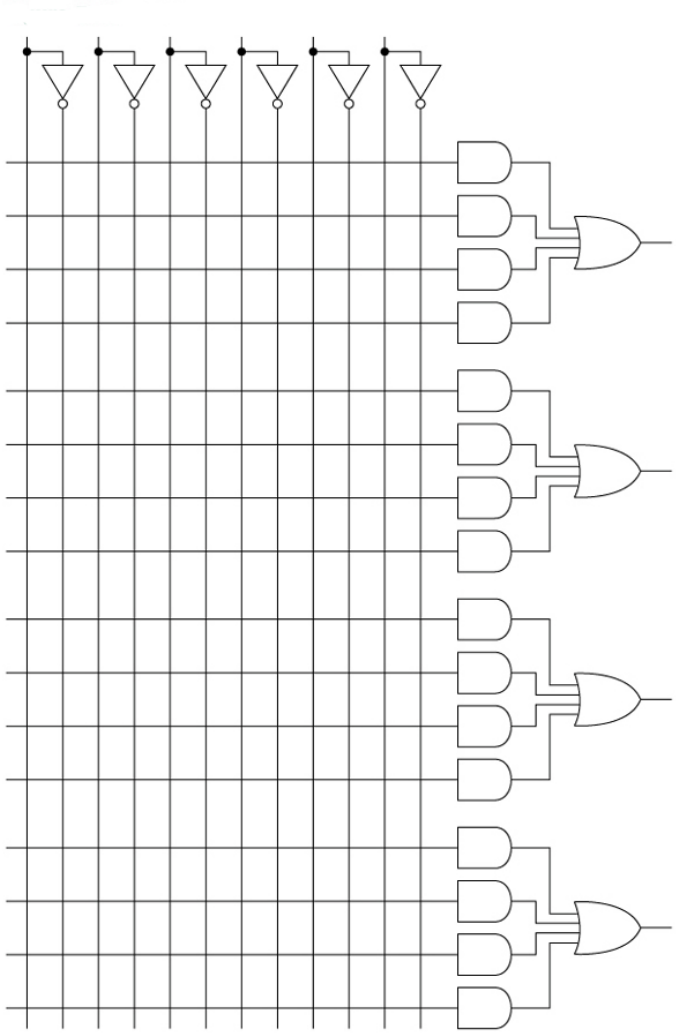
Y = A'C'D' + ACD + BCD



PALs

- Programmable Array Logic (PAL) chips implement a number of Sum-of-Product expressions
- PALs are specified by
  - the number of inputs (variables)
  - the number of product terms (AND gates)
  - the number of functions (OR gates)
- Every input variable (or its complement) may be included in a product term by setting a programmable switching element.
- Every function (output) has a **fixed** number of product terms.

PAL Structure



PAL Example

W = AB'C' + CD

X = A'BC' + ACD' + A'CD + {BCD or ABC}

Y = A'C'D + ACD + {A'BD or BCD}