Chapter 13

IMPLEMENTATION OF COMBINATIONAL LOGIC BY PROGRAMMABLE LOGIC DEVICES
Lesson 3

Programmable Array Logic
PAL Circuit

AND End $2 \times n \times l$ Fusible Links ($l < 2^n$). $i / 0$ and $j$ [$(2^n - 1)$]. Number of ANDs = $l =$ number of miniterms provided for in PAL
Programmable Array Logic (PAL)

- A special programmable logic device (PLD) in which each input of the AND gate in the AND-OR array have a fusible link, which is fused as per the truth table or the Boolean expressions or Karnaugh map for each output. [OR array inputs are not programmable in PAL.]
Input to ANDs and ORs

- \((l \times m)\) PALs have \(n\) input AND gates \((l\) in number\) and \(l\) input \(m\) OR gates.
- \(l < 2^n\) maximum possible ANDs due to \(2^n\) maximum possible miniterms in a \(n\)-variable Boolean function.
- Each AND output corresponds to a mini
  term each in the SOP expression.
- Total Number of fusible and programmable
  links = \(2 \times n \times l\) only in place of \((2 \times n \times 2^n \times m\) maximum possible\).
PAL lesser number of ANDs = \( l \)
compared to \( 2^n \) in a PROM

- Assume that \( n \) = quite high, say 10
- Computer based minimization procedures available,
- Makes it feasible to implement a combination logic circuit with fewer prime implicants only. [For example, \( m' \) prime implicants, where \( m' < 2^n \) for \( n \)-variable functions]
PAL lesser number of ANDs compared to $2^n$ in a PROM

- Occurrence of common sets and complementary sets of prime implements in the Boolean expressions reduces the requirements of ANDs. So the PAL will have less cost (less number of gates) requirement and $l < 2^n$. 
OR and AND Gates

- An OR gate with no input given at one of its line is assumed to have input 0.
- An AND gate with no input given at one of its line is assumed to have input 1.
Diagrammatic Representation

- Total lines from \( l \) ANDs, each with no fusible link to each OR can be shown as a single vertical line with the \((l \times n)\) fixed connections.

- Total \( 2 \times n \) lines to each AND has fusible links in PAL and can be shown as a single horizontal line with \( 2 \times n \) fusible links.
Diagrammatic Representation

- Total number of vertical lines = number of OR gates = number of Boolean functions, which are implemented
- Total number of ANDs horizontal lines = $l = \text{number of miniterms, which can be used for implementation by programming } (2 \times n \times l) \text{ fusible links}$
Fusible Link Diagrammatic Representation

- A fusible link, which is not fused (snapped) is shown by a dot sign at an intersection at the input from a previous stage output or from a source of input to the link. A cross or a missing sign is equivalent of unsoldered link in the electrical circuit.
Alternative Representation

• A fusible link, which is fused (not snapped) is shown by a cross sign at an intersection at the input from a previous stage output or from a source of input to the link. A cross sign is equivalent of soldered link in the electrical circuit.
Alternative Representation

• A fusible link that is fused (snapped) is shown by a missing cross sign at the intersections at the inputs from the previous stage outputs. A missing cross sign is equivalent of un-soldered (detached) link in the electrical circuit.
PAL with provision for feedbacks from ORS

- \( n = (n' + k) \), where \( n' \) = number of literal input variables in Boolean functions and \( k \) number of feedback inputs (among \( m \) ORs for the outputs)
- 16L8 is a PAL, which has 10 input variables and 6 OR feedback variables from 6 of the 8 ORs.
16L8

- Between OR-array output and the output pin at the PAL IC, for each OR there is a tristate NOT. This means that the output pins are active ‘0’. An output pin represents Boolean function $Y = 1$ when pin is at ‘0’ and in tristate (inactive) state, when $Y = 0$. 
16L8 PAL

- Output of 6 ORs can be feedback at the ANDs with fusible links
- Two OR outputs do not have feedback and their outputs are called dedicated outputs.
16L8 PAL

- Number of literal variables of Boolean functions $= n = 10$. Number of OR feedbacks $= m' = 6$. Maximum $(n + m) = 16$ inputs in 16L8. 16 inputs can be the Boolean variables are thus variables or feedback from the ORs.

- Number of inputs at one horizontal line to one AND is 32. [16 + 16 complements]
16L8 PAL

- 32 inputs (16 Inputs and 16 complement of them) per AND links 7 AND miniterms plus one AND.
- 8 ANDs in array.
- 8 ORs, each with 7 miniterm inputs
16L8 PAL

- Each AND is a 32-input AND.
- 8 ANDs $\times$ 8 ORs = 64 rows $\times$ columns.
- Maximum number of links, therefore, 2048 (= 32 $\times$ 64) out of which (= 32 $\times$ 8) at the ANDs are programmable by an external unit called PAL programmer.
16L8 PAL

- \( l = 7 \) from miniterms to OR + 1 due to additional AND
- Eight ANDs, but each OR has only 7 input from miniterms.
- \( m = 8 \) ORs for maximum 8 outputs.
Registered PAL

- It is a PAL, which registers the outputs when a clock input is applied and in presence of clock input only, the outputs then change according to the inputs at the ANDs. After the clock input is withdrawn, the outputs don't change even if the inputs are withdrawn or changed.
Registered PAL

• It is a PAL used for sequential circuits
PAL programmer

- A programming is a systemic hardware related procedure implemented by software at a programming system, called programmer or laboratory programmer or PLD programmer and the programming means executing the procedure for fusing (snapping) the needed links in the erased or fresh PAL.
Erased PAL

- Erased or fresh PAL means PAL in which fusible links ready for programming
Summary
• Implementation of Boolean Functions by PAL — PAL implements Boolean functions of combinational circuits by programming the AND gate links, output of which connects to the ORs.
(n × m) PAL has 2n input AND gates (l < 2^n in number) with fusible links and l input m OR gates each with no fusible links.

Number of fusible links L are 2 × n × l only in place of (2 × n × 2^n × m) maximum possible in AND-OR array with no feedbacks.

n = (n’ + k), where n’ = number of literal input variables in Boolean functions and k number of feedback inputs (among m ORs)
End of Lesson 3 on
Programmable Array Logic
THANK YOU