Control Unit Implementation

Lookup Tables can consume large amounts of storage

Using a full 10-bit address (opcode + state) means 1024 entries

Each entry contains 16 control bits and 4 next state bits

Total size = 1024 * 20 bits (expensive in earlier days of computing)

One way to reduce the required storage is to use a PLA

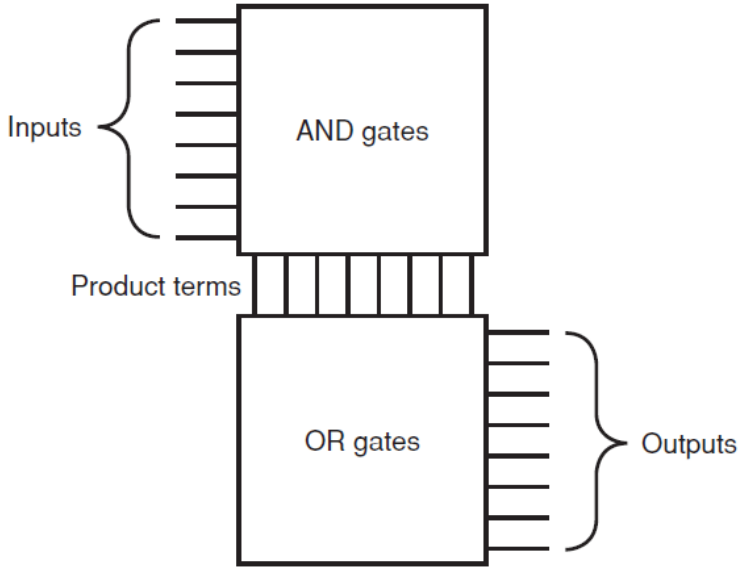
"PLA" means programmable logic array

PLA Implementation

Each PLA output is a logical sum of one or more minterms

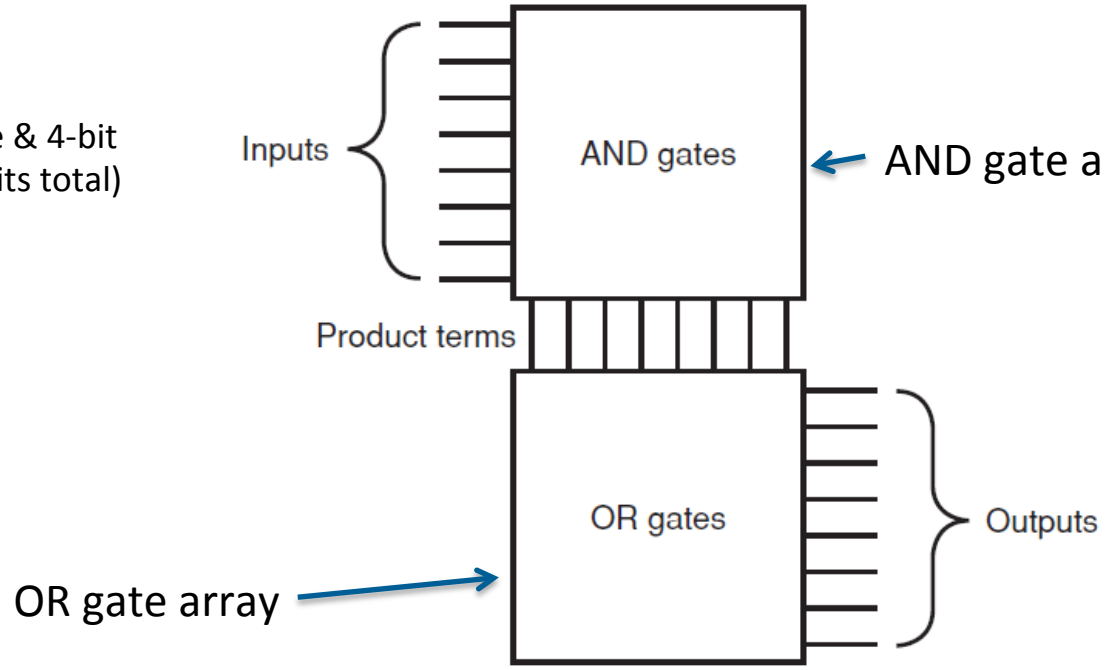
Minterm (or product term) is a logical AND of two or more inputs

A minterm corresponds to a single row in a truth table



MIPS Core Subset PLA

6-bit Opcode & 4-bit state# (10 bits total)



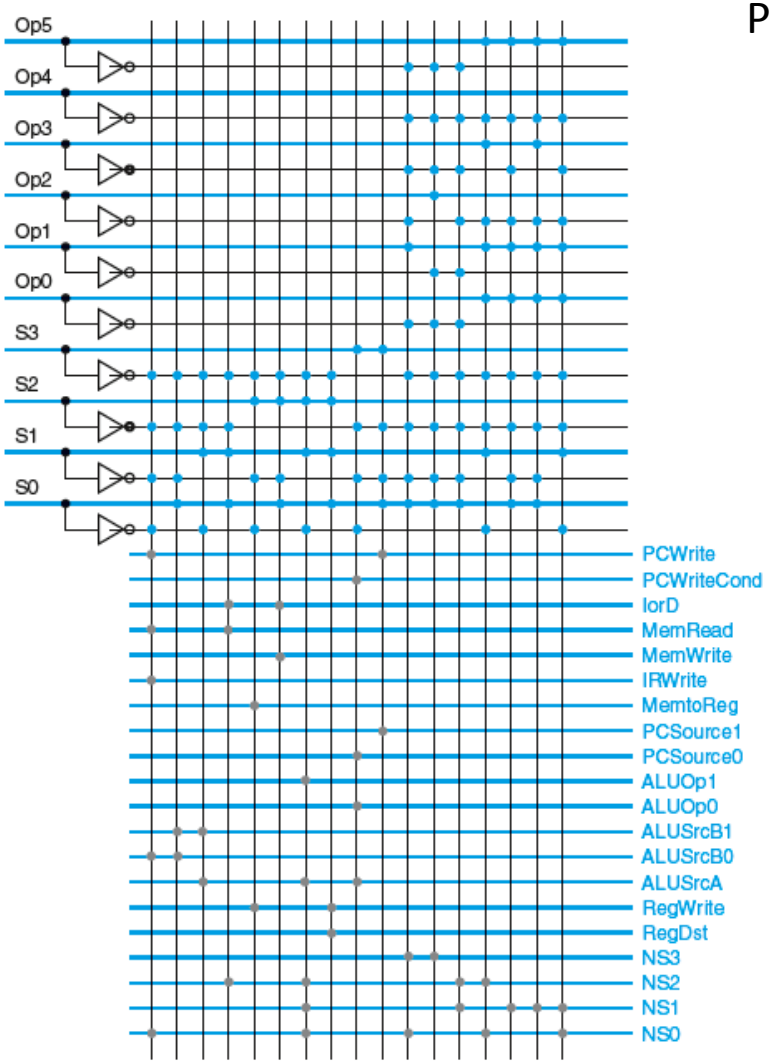
16 Control bits & 4-bit next state (20 bits)

- AND array generates products of inputs or inverted inputs
- OR array generates logical sums of product terms

PLA Example

Programmable Logic Array

One vertical line for each of the 17 minterms

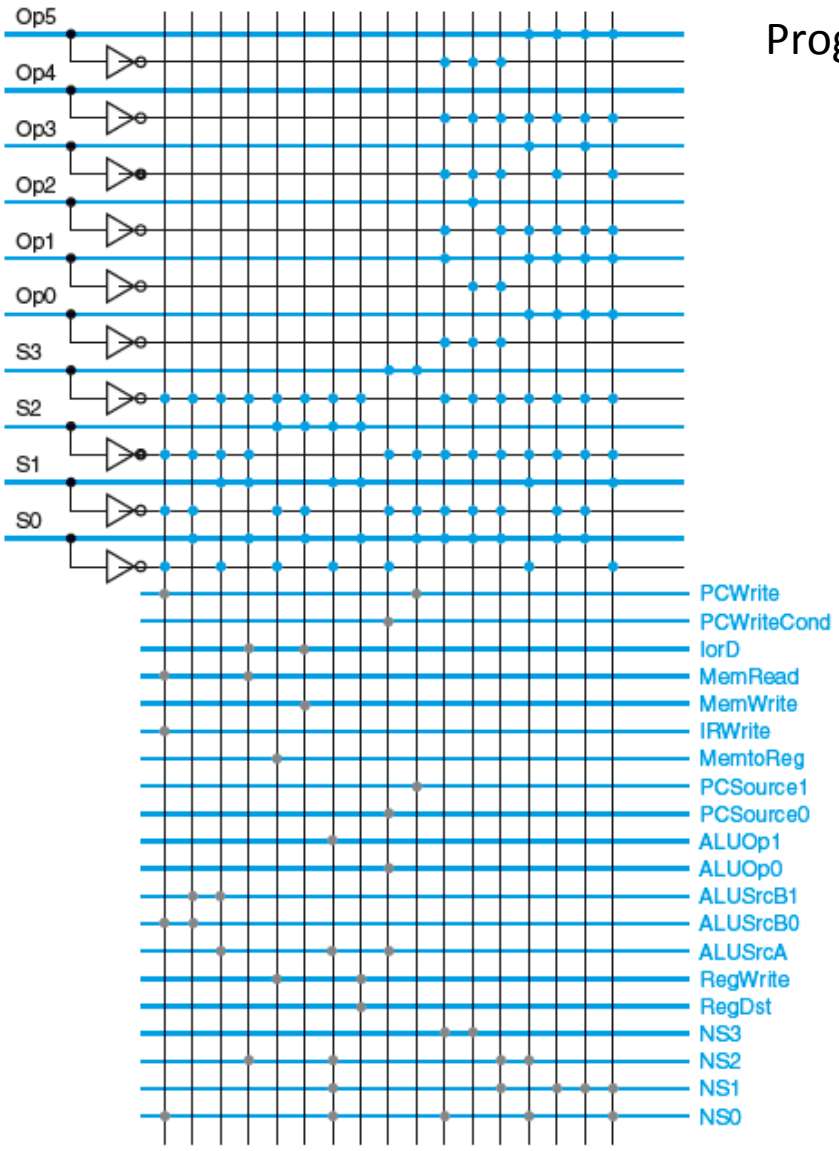


Size proportional to (#inputs + #outputs) * #product_terms

Horizontal output lines correspond to control lines and next state bits

PLA Example

Programmable Logic Array



The 10 Inputs are opcode and current state

Size proportional to (10 + 20) * 17 = 510

The 20 outputs are control signals and next state

PLA Implementation

Opcode bits and current state# and fed in for each cycle

The PLA outputs the resulting control signals and next state#

Control signals are output for each instruction step or subcycle

The sequence of state numbers identify the steps or transitions

This PLA supports our MIPS core instruction subset

PLA Implementation

Each vertical line in the previous PLA is a minterm

The leftmost 10 depend only on the state

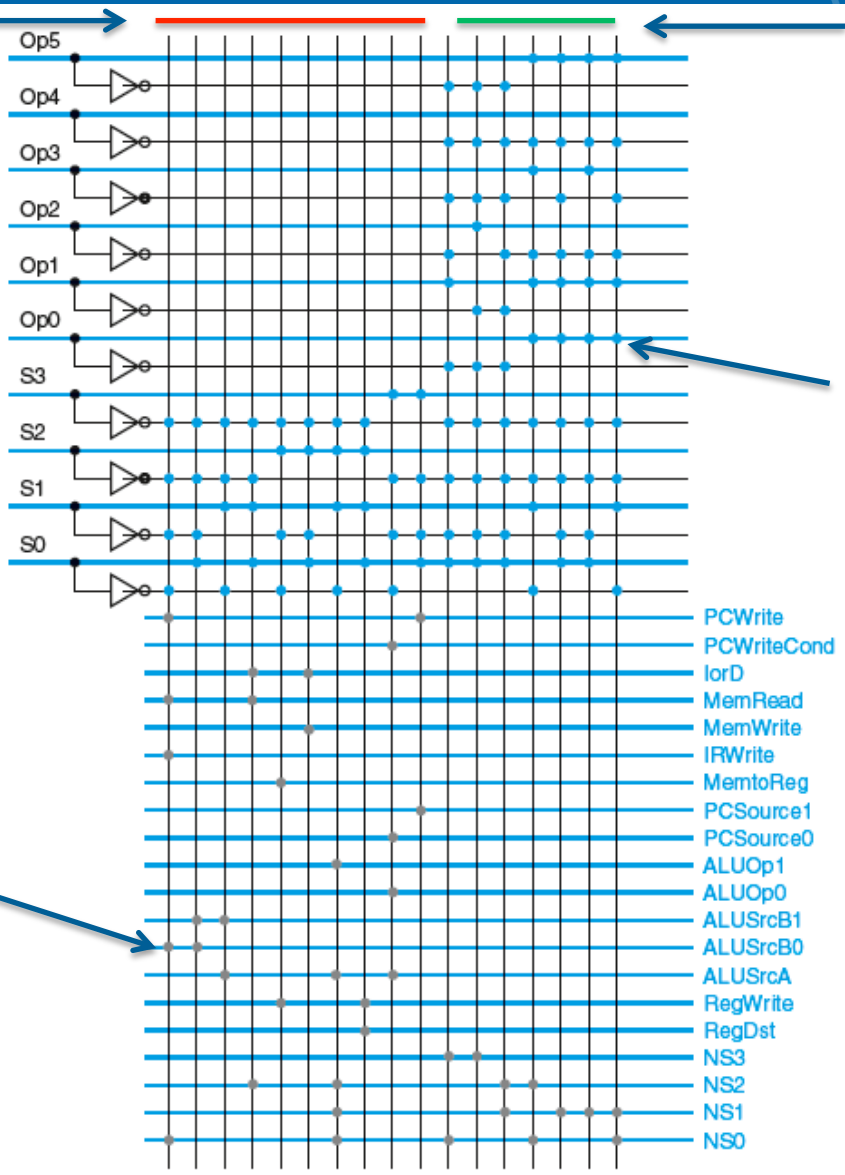
The remaining 7 depend on the state and opcode

The top half of the figure is the AND plane that computes minterms
Dots in top half indicate which inputs are fed into AND gate

The bottom half is the OR plane that sums the minterms
Dots in lower half indicate which minterms are fed into OR gate

PLA Implementation

These 10 minterms depend only on state#



These 7 minterms depend on opcode as well as state#

Blue dots identify AND gate inputs

Black dots identify OR gate inputs

OR gates produce the sum of products

PLA Implementation

Previous PLA can be replaced by two smaller PLAs (PLA1 & PLA2)

PLA1

- produces 10 minterms from 4 inputs (state)
- outputs the 16 control signal bits
- Size = (4 + 16) * 10 = 200

PLA2

- produces 7 minterms from 10 inputs (opcode & state)
- outputs 4-bit next state number
- Size = (10 + 4) * 7 = 98

Together the two consume less space than the single larger PLA