Lecture 4 – Programmable Logic Devices

Karl R. Wilcox
Karl@cs.rhul.ac.uk

Administration

- There will NOT be a lecture or Lab session this Wednesday
- However, the TkGate package has been updated on Aspasia and is available for use.
- Lecture Notes
  - Are available at http://www.cs.rhul.ac.uk/~karl
  - Formats
    - Original Powerpoint 2000 slides
    - PDF files, A4 2 slides per page, colour
    - Let me know if you would like other formats

Objectives

- **In this lecture we will discuss**
  - **The story so far…**
  - **State Machine Implementation Using PALs**
  - **Other Programmable Logic Devices**

Review

- We have looked at State Machines
  - As generally useful tools
- We have looked at implementing state machines
  - Using ROMs
- Problems with ROMS
  - Inefficient use of logic – can't take advantage of "don't care" input combinations
  - ROM architecture is inflexible – fixed number of inputs and outputs

Truth Tables

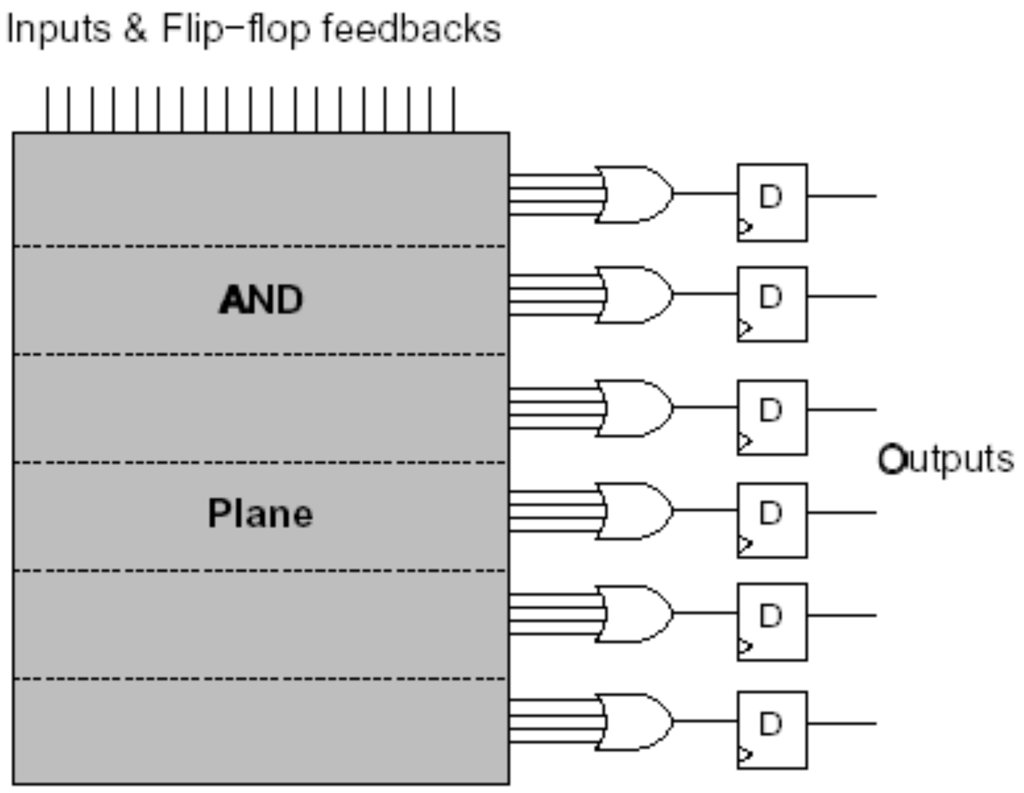
- **To populate the ROM we had to construct a truth table**
- **Rows of the truth tables can be expressed as *minterms***
- **Any function can be re-written as the sum of its (true valued) minterms**
  - Known as *Canonical Sum of Products* Form

Minterms

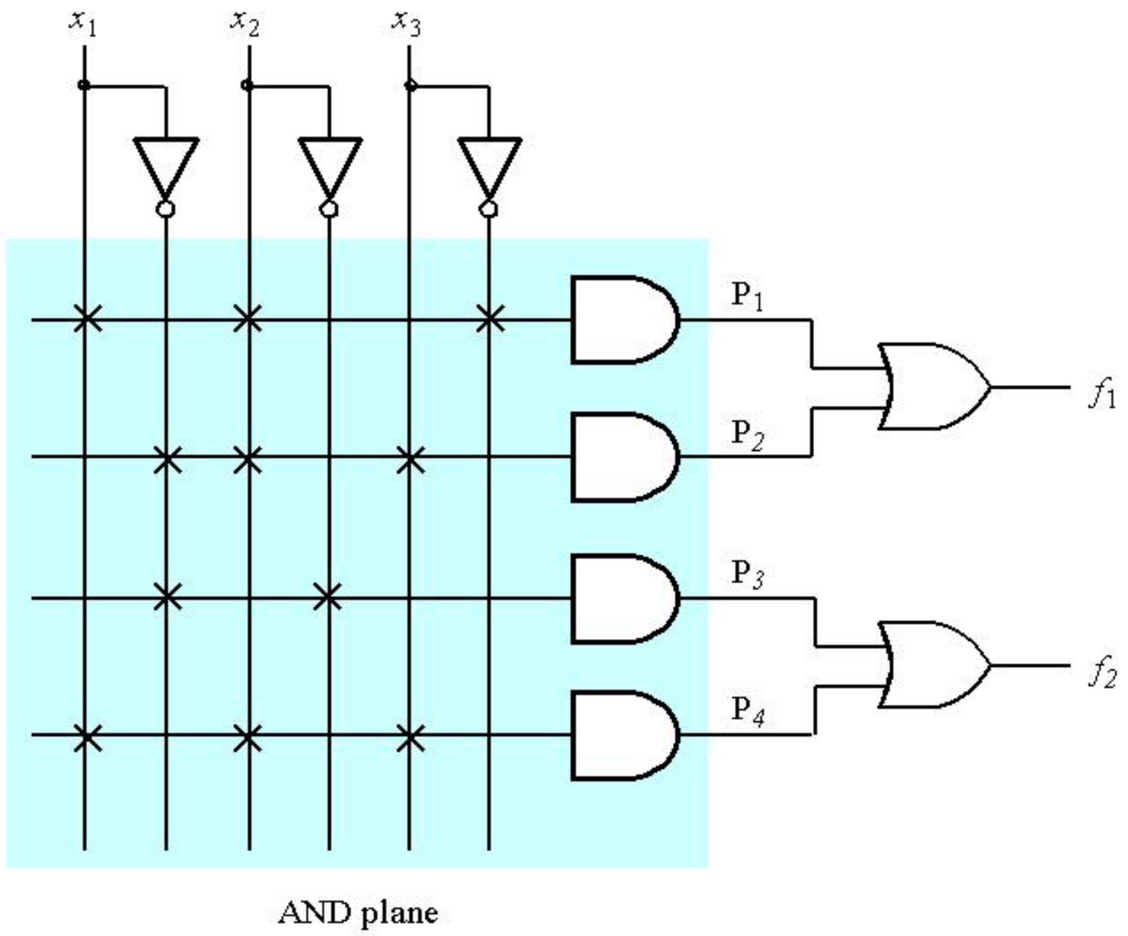
Minterm No.	Minterms			c	b	a	f
0	c’	b’	a’	0	0	0	0
1	c’	b’	a	0	0	1	1
2	c’	b	a’	0	1	0	1
3	c’	b	a	0	1	1	0
4	c	b’	a’	1	0	0	1
5	c	b’	a	1	0	1	0
6	c	b	a’	1	1	0	0
7	c	b	a	1	1	1	1

$$f = (c'\cdot b' \cdot a \;+\; c' \cdot b \cdot a' \;+\; c \cdot b' \cdot a' \;+\; c \cdot b \cdot a) \;=\; \Sigma(1,2,4,7)$$

Programmed Array Logic (PAL)



PAL Detail

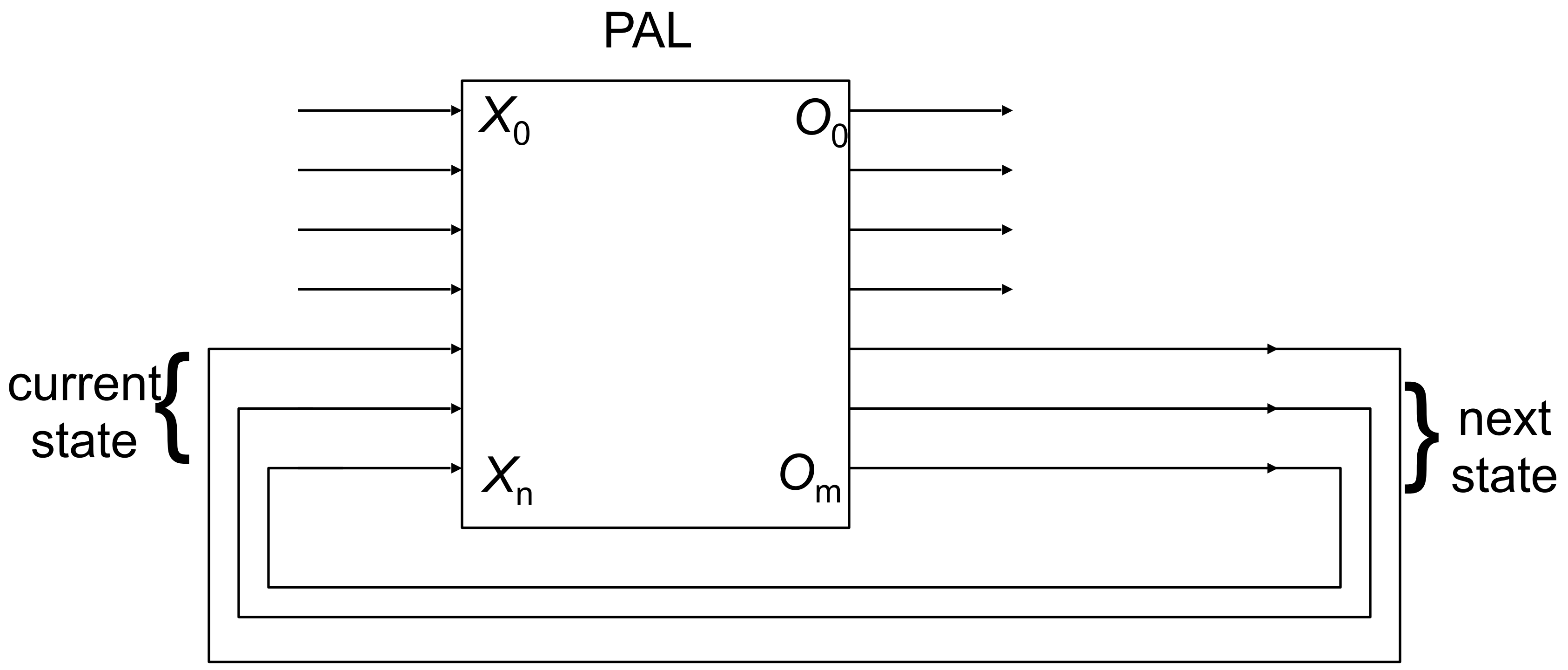


PAL Features

- A programmable AND plane
  - Usually by fuses (hence one-time programmable)
- A fixed OR plane
- Outputs routed via flip-flops
- Available in various configurations
  - Different numbers of inputs and outputs
  - Various sizes of OR gates

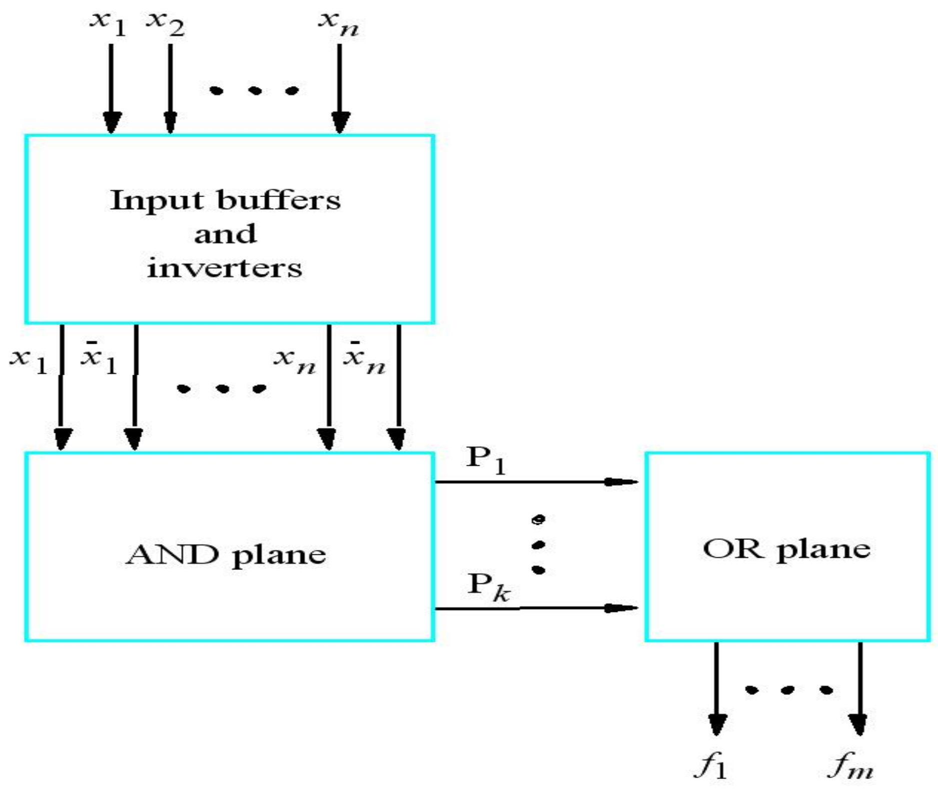
PAL State Machines

- **State Machine Implementation using a PAL**

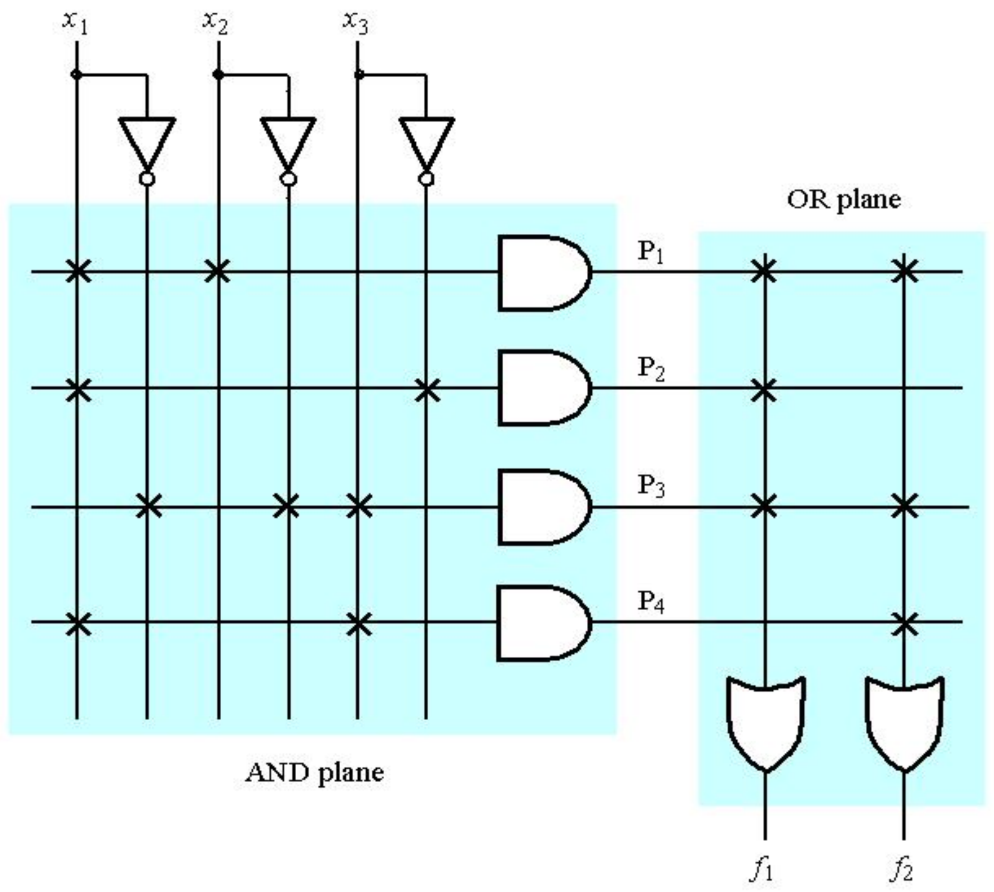


Programmable Logic Array (PLA)

- **Development of the PAL to include a programmable OR plane**


x_1 x_2 x_n
Input buffers and inverters
x_1 $\bar{x}_1$ x_n $\bar{x}_n$
AND plane
P_1
P_k
OR plane
f_1 f_m

PLA in Detail



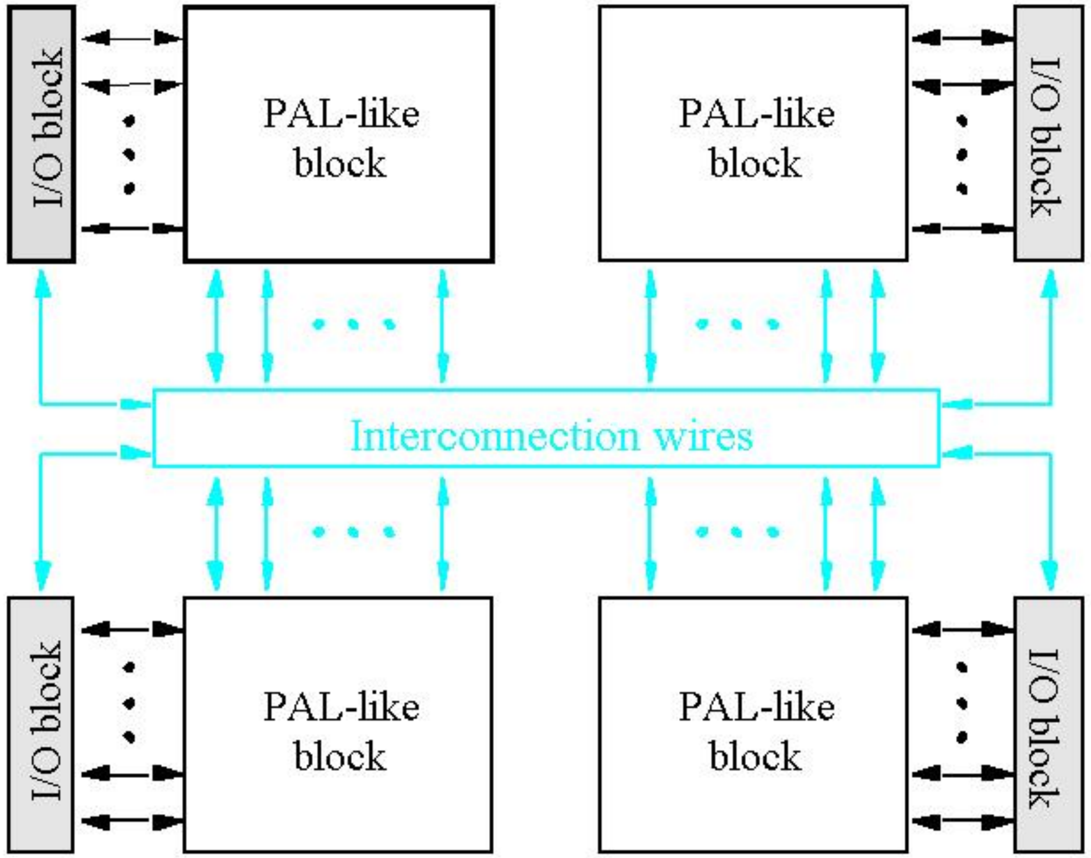
PLA Features

- As with PAL, any inputs or complements can be AND'd together in the AND plane
- Any of these products can be OR'd in the OR plane
- Typical devices:
  - 16R8: Up to 16 inputs (8 fixed, 8 input/outputs), 8 "registered" outputs (D type flip-flops)
  - 22V10: 22 inputs, 10 outputs, versatile outputs (registered or not, as required)
- Cheap, fast logic implementation
- Very widely used
- Need automated support to configure logic arrays

SPLDs and CPLDs

- **Programmable ROMs (PROMs), PALs and PLAs are closely related**
- **Generically known as Simple Programmable Logic Devices (SPLDs)**
- **Later developed into Complex Programmable Logic Devices (CPLDs)**
  - **Combines multiple SPLDs with programmable interconnections**

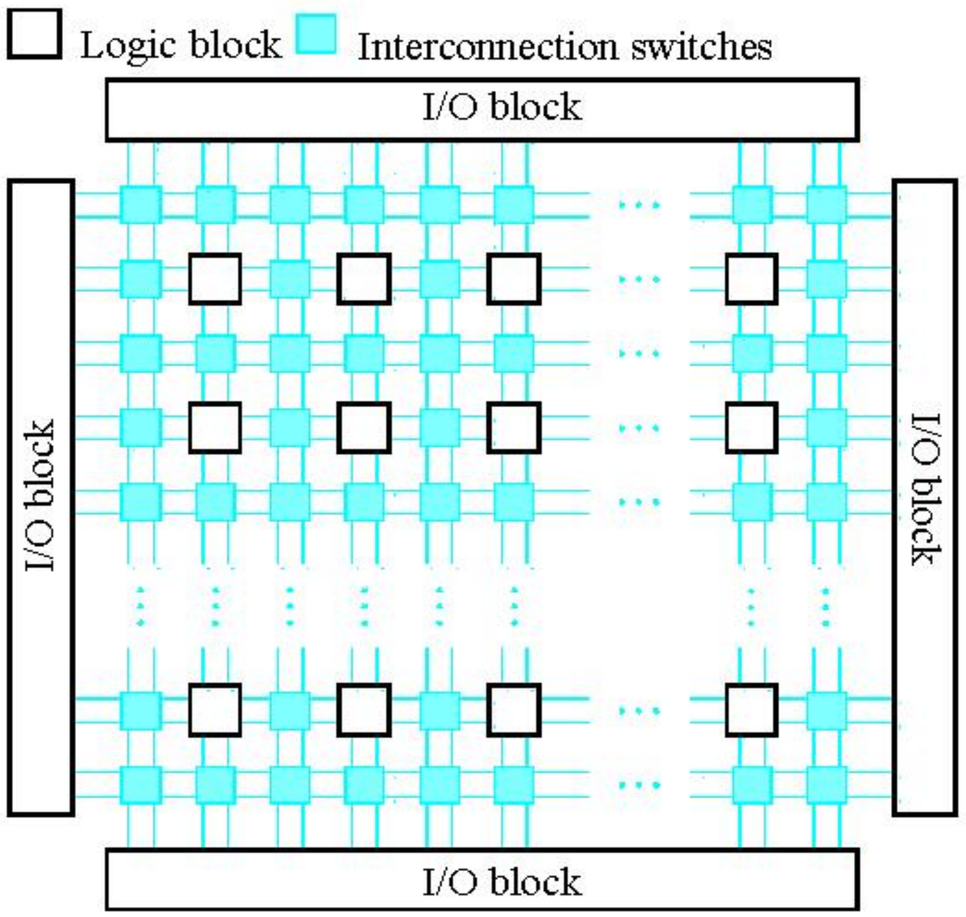
CPLD Structure



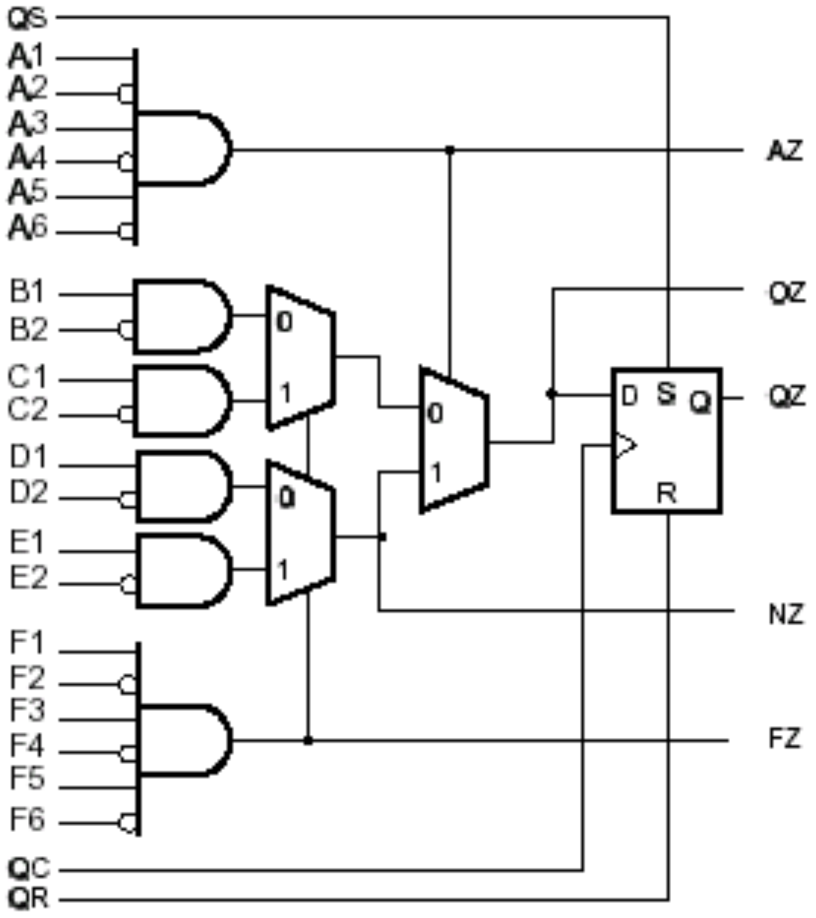
CPLD Applications

- Can construct reasonably complex designs
- Graphics controllers, LAN controllers, bus interface logic
- All CPLDs are re-programmable
  - Memory controlled switches, anti-fuses
- Some are re-programmable in circuit
  - E.g. modems allow new communication protocols to be downloaded

Field Programmable Gate Arrays



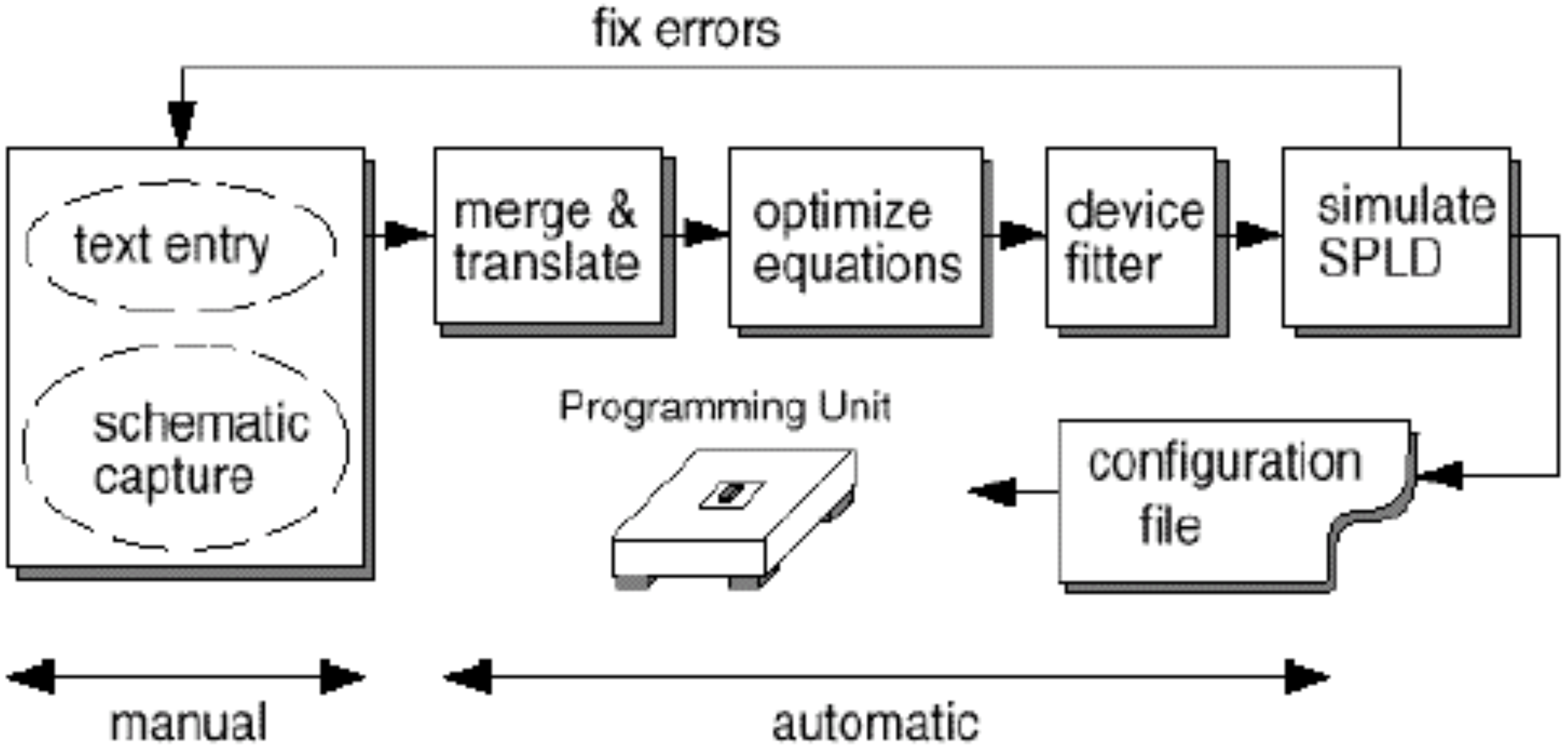
FPGA Logic Block



- Wide variation in logic block functionality, this is an example only
- Needs a high level of CAD support

FPGA CAD Design Process

- Design languages include ABEL (simple), VHDL



FPGA Applications

- Device controllers, random logic, communications controllers, custom CPUs
- Prototyping designs to be implemented as Mask Programmable Gate Arrays
  - Semi-custom logic chips
  - Final metal layer determines logic function and interconnections
- Multiple FPGAs can simulate entire hardware systems for prototyping

Summary

- State Machines can be implemented using PALs
- PALs, ROMs and PLAs are all examples of SPLDs
- CPLDs incorporate multiple SPLDs with programmable interconnects
- FPGAs have a large number of logic blocks, interconnects and I/O blocks
- All need increasing levels of CAD support

Next Week

- A more general minimisation method than Karnaugh maps
  - Quine-McCluskey Minimisation

- Next Lecture, Monday, 2nd Feb, CS C103