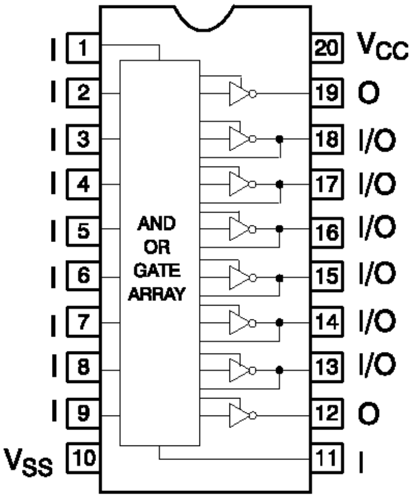
PALs and PLDs

Programmable logic devices first came on the scene around 1980 with the Programmable Array Logic (PAL) form Monolithic Memory, Inc. (MMI) which had similarities to the Programmable Logic Array (PLA) traditionally found in Application Specific Integrated Circuits (ASICs). This was a fuse-based programming technology which was touted to reduce PCB manufacturing inventory as a result of the programmability – the interesting thing was that the PAL databook was as thick as the TTL data book with many, many different versions of PALs. However, only two of the PALs proved to be high volume devices that could actually reduce manufacturing inventory: the 16L8 and the 16R8 (see subsequent pages for architectural details).

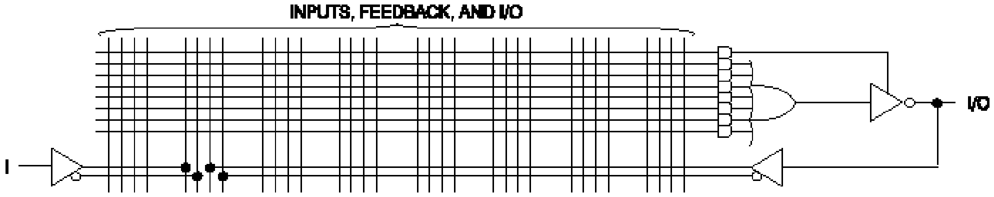
As other manufactures of programmable logic came on the scene, a more generic term was sought since PAL was a registered trademark of MMI and hence PLD became the generic term for Programmable Logic Devices. With the new manufacturers came new and more generic architectures like the 16V8 which was a hybrid of the 16L8 and 16R8 as well as the 22V10 which had larger product terms and more outputs than the 16V8. Both of these devices introduced the “macrocell” to facilitate implementation of user defined combinational or sequential logic on a per output basis.

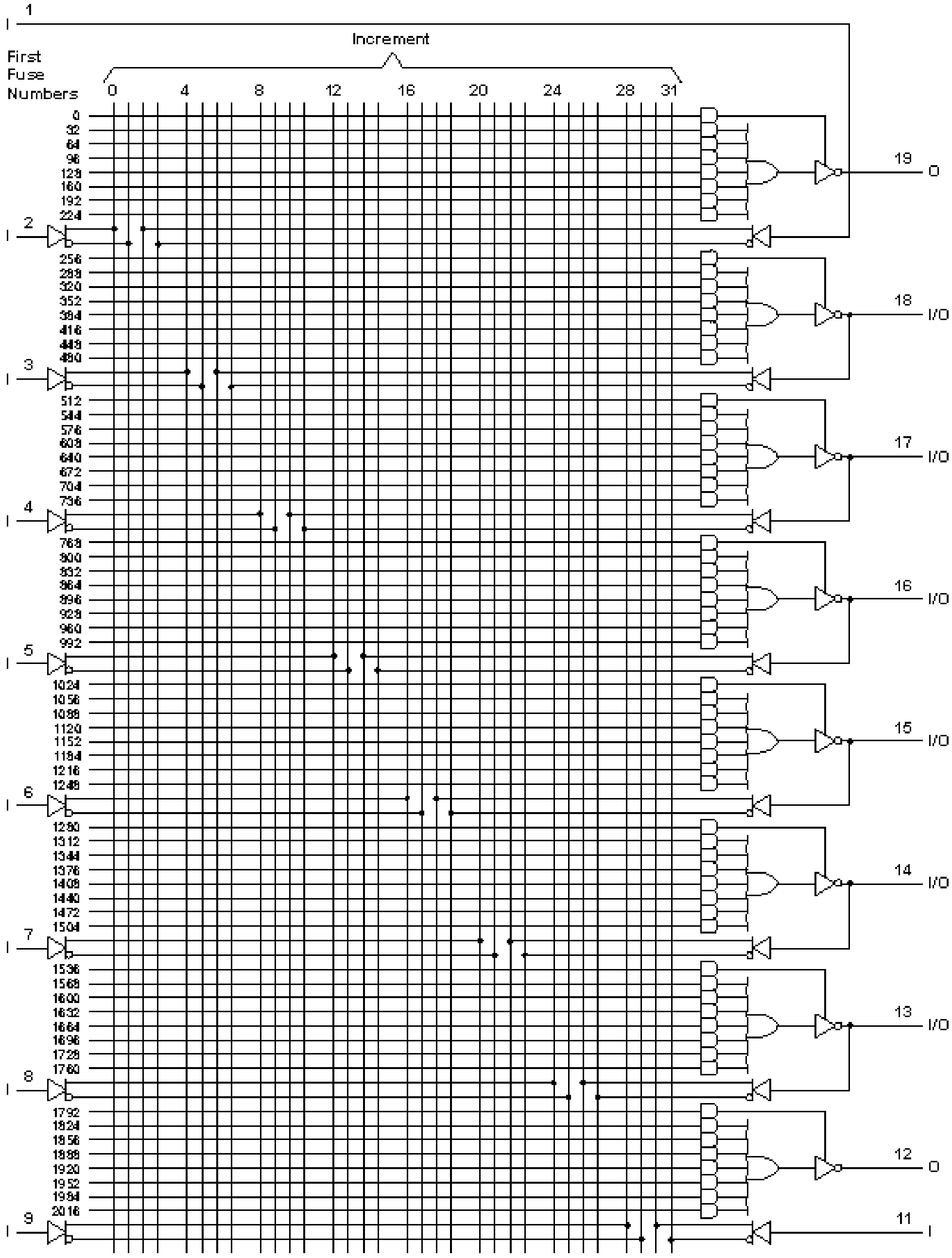
Note that none of these devices included product term sharing as was the case in PLAs. During the early years of growth in PLD architectures, the number of macrocells was increased and new features such as product term sharing between adjacent macrocells and product term stealing (reallocating product terms from one macrocell to an adjacent macrocell) were introduced. The advent of the Complex Programmable Logic Device introduced the concept of an array of PLDs with a programmable routing network, sometimes referred to as a Programmable Interconnect Module (PIM). As CPLDs grew in size, they began to look more like FPGAs but with larger Programmable Logic Blocks until about the only difference between FPGAs and CPLDs was that the combinational logic in CPLDs was PLA-based while the combinational logic in FPGAs was based on Look-up Tables (LUTs). Today, the term FPGA generally refers to all programmable logic.

16L8 PAL – Combinational Logic



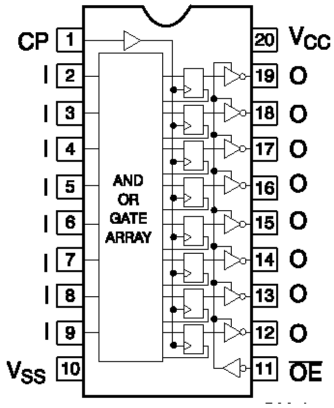
Up to 16 inputs (10 dedicated inputs only)
 32 bit & bit-bar lines
Up to 8 outputs (2 dedicated outputs only)
Up to 7 product terms per output
1 product term/output for tri-state control
 Input, Output, Bi-driectional bus (on per output basis)
Note fuse numbers (early technology)





Fuse number = First fuse number + Increment

16R8 PAL – Sequential Logic



16 inputs (counting feedback into array from DFFs)

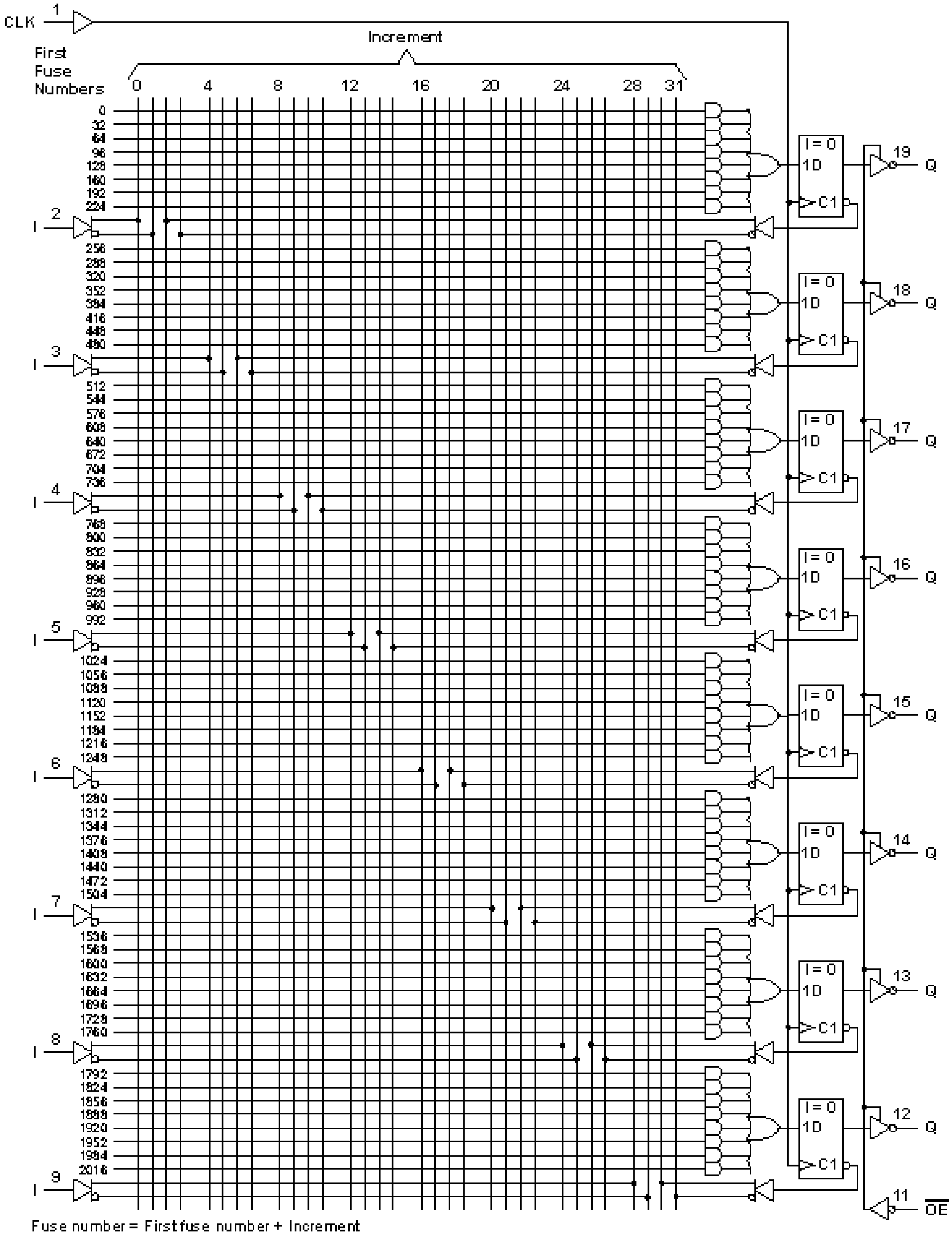
Again 32 bit & bit-bar lines

8 outputs (Q outputs from 8 DFFs)

Up to 8 product terms per FF input

Common tri-state control from common input pin

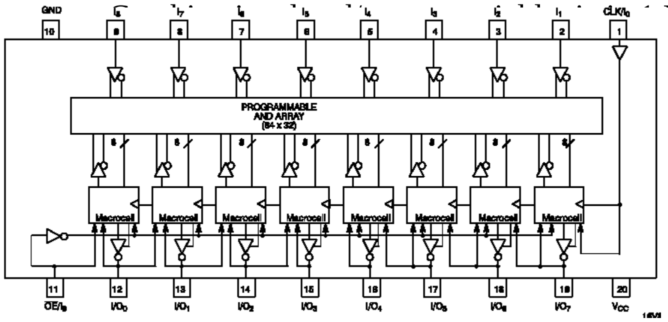
Output or tri-state bus only (not bidirectional)

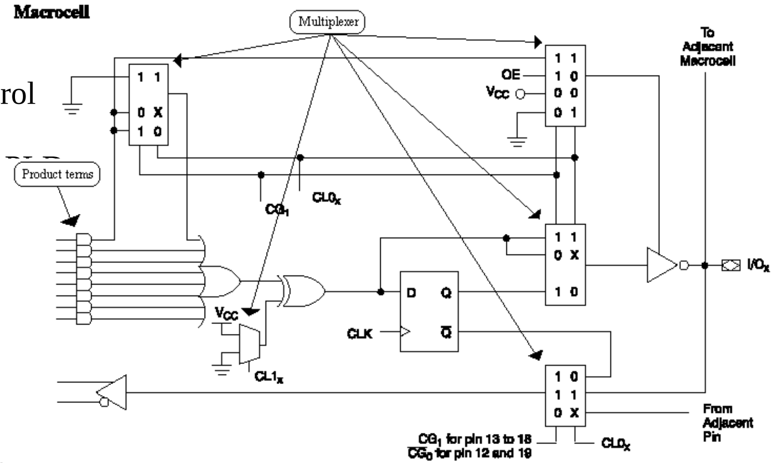


Fuse number = First fuse number + Increment

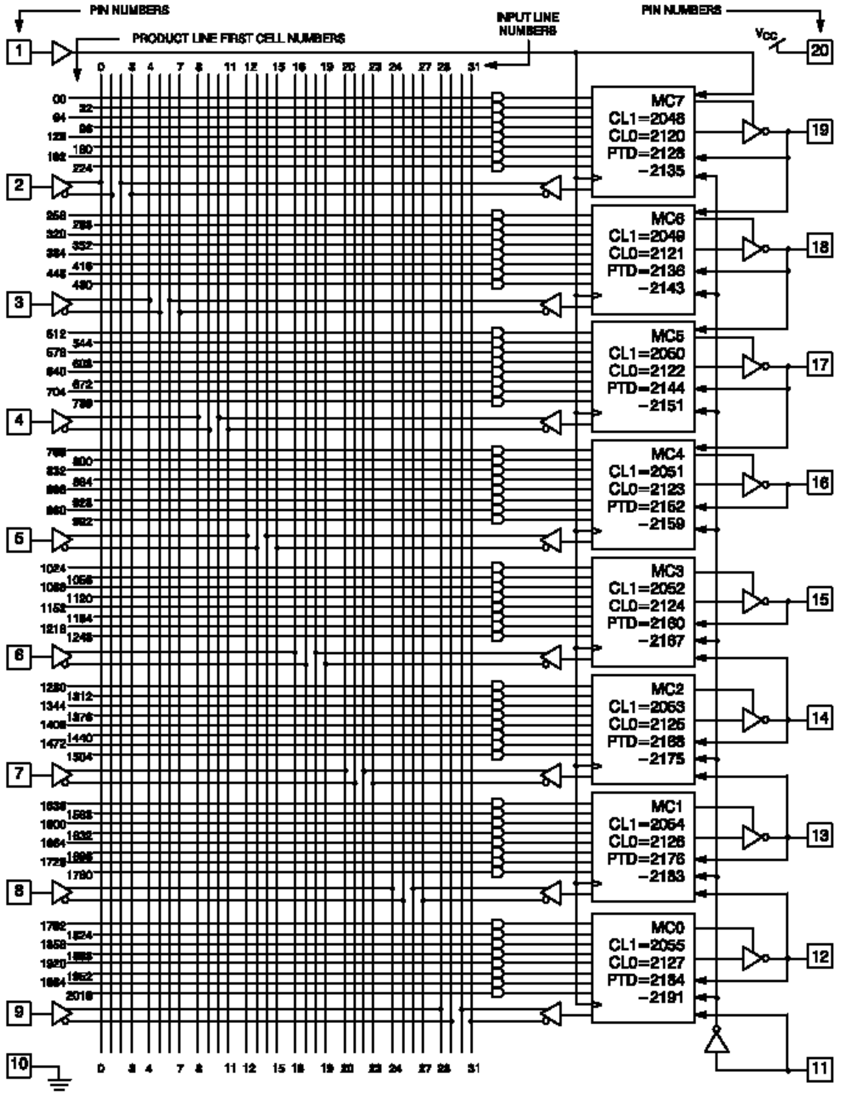
PLD 16V8 – Combinational/Sequential Logic

- Up to 16 inputs (8 dedicated inputs only)
  - 32 bit & bit-bar lines
- Up to 8 outputs
- Up to 8 product terms per output
  - optional product term/output for tri-state control
- Introduction of Macrocell



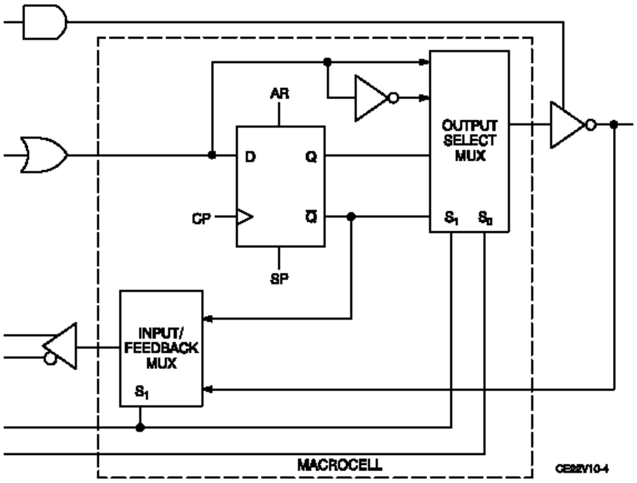


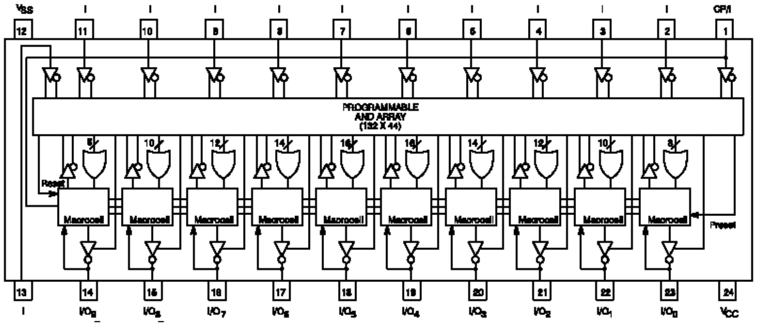
CG_0	CG_1	$CL0_x$	Cell Configuration	Devices Emulated
0	1	0	Registered Output	Registered Med PALs
0	1	1	Combinatorial I/O	Registered Med PALs
1	0	0	Combinatorial Output	Small PALs
1	0	1	Input	Small PALs
1	1	1	Combinatorial I/O	16L8 only

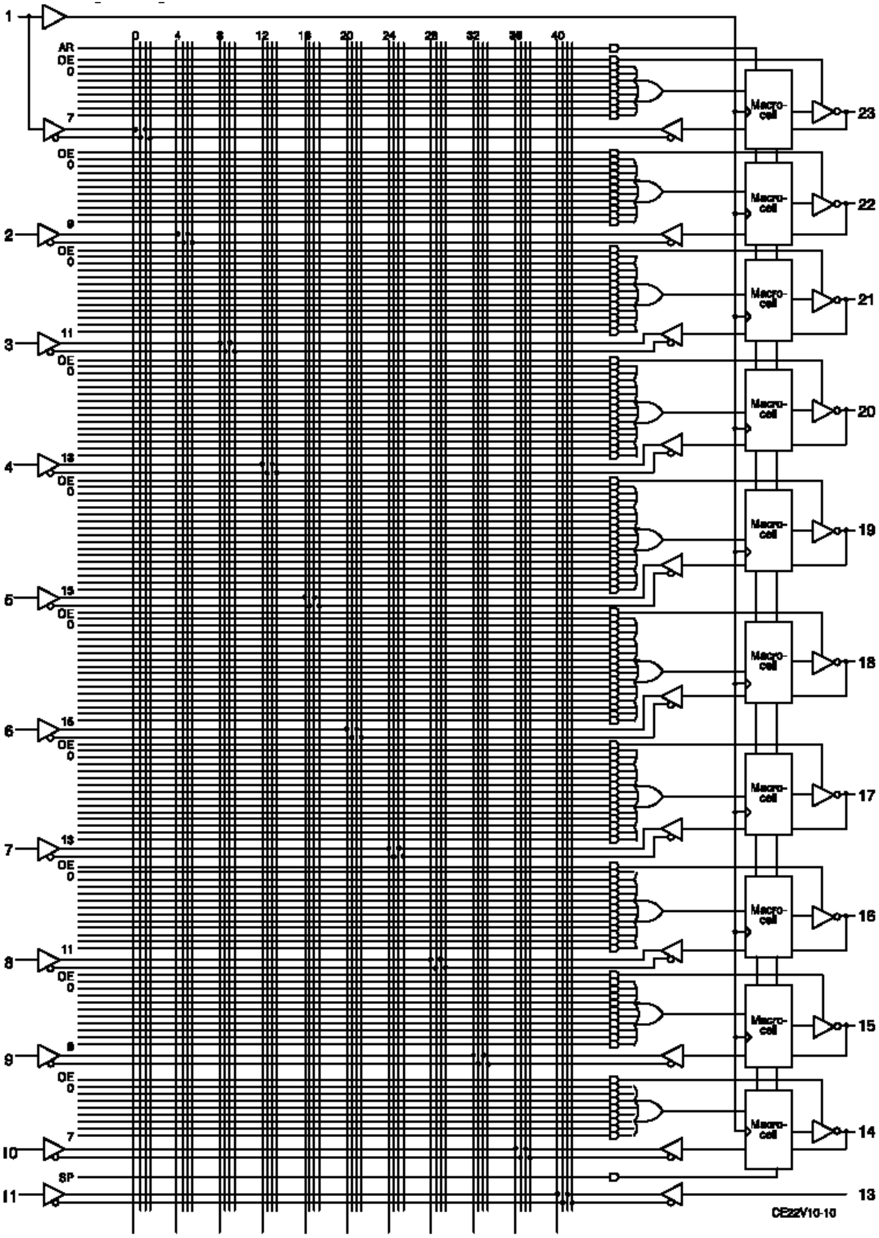


PLD 22V10 – Combinational/Sequential Logic

- Up to 16 inputs (8 dedicated inputs only)
  - 32 bit & bit-bar lines
- Up to 8 outputs
- From 8 to 16 product terms per output
  - dedicated product term/output for tri-state control
- Introduction of Macrocell
  - Combinational and/or sequential logic in 1 PLD







Configuration Table

Registered/Combinatorial		
C_1	C_0	Configuration
0	0	Registered/Active LOW
0	1	Registered/Active HIGH
1	0	Combinatorial/Active LOW
1	1	Combinatorial/Active HIGH