

# Fundamentals of Computer Systems

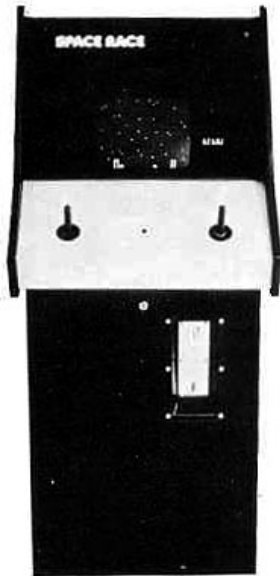
## Memory

Stephen A. Edwards

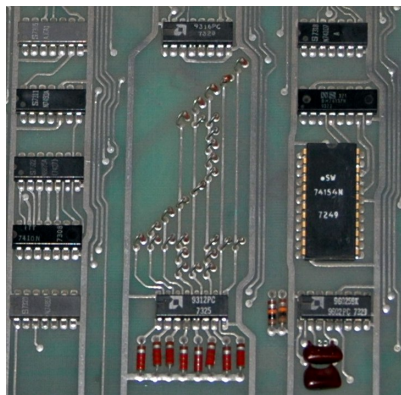
Columbia University

Summer 2017

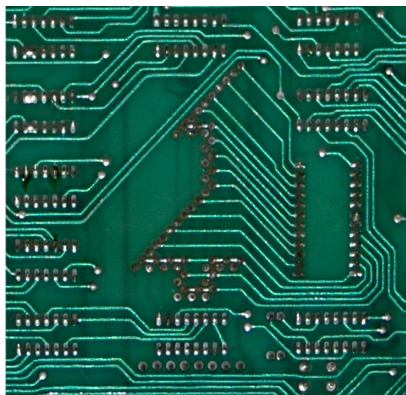
# Atari Space Race, 1973



# Atari Space Race PCB

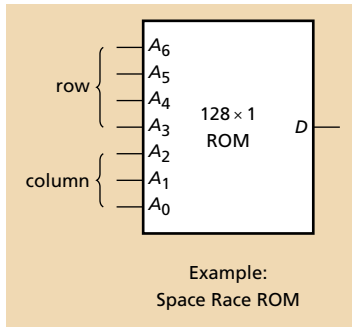
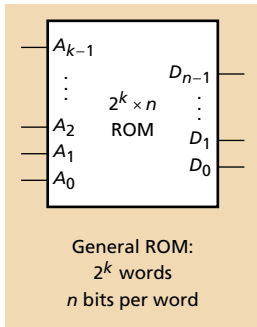


Front

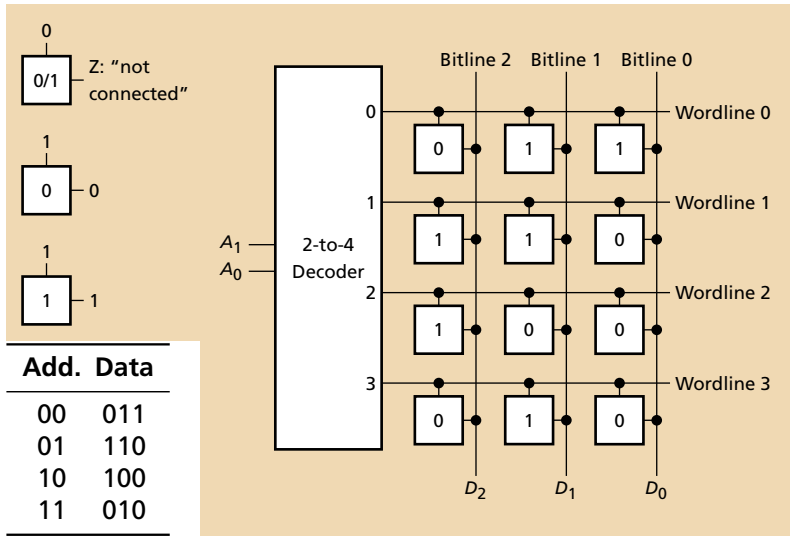


Back (mirrored)

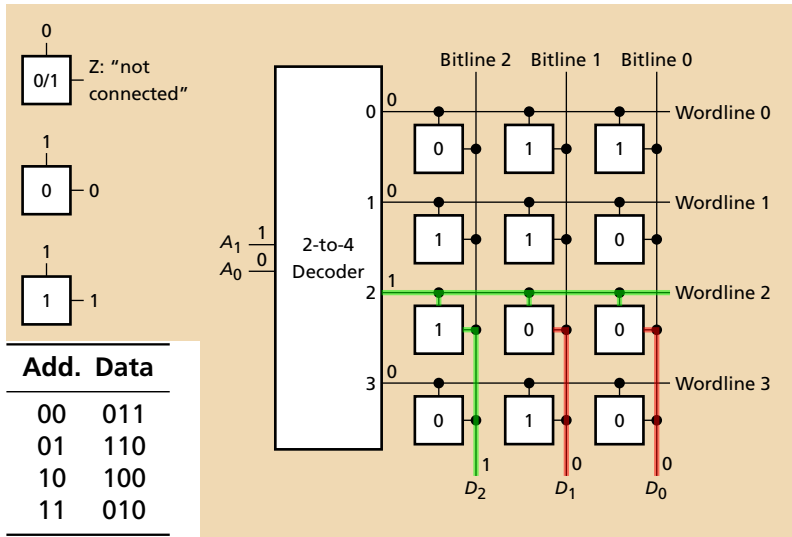
# Read-Only Memories: Combinational Functions



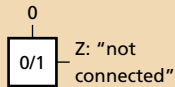
# Implementing ROMs



# Implementing ROMs

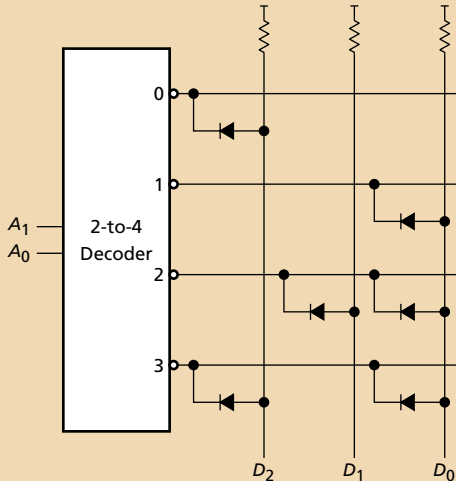


# Implementing ROMs

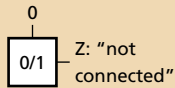


## Add. Data

00	011
01	110
10	100
11	010

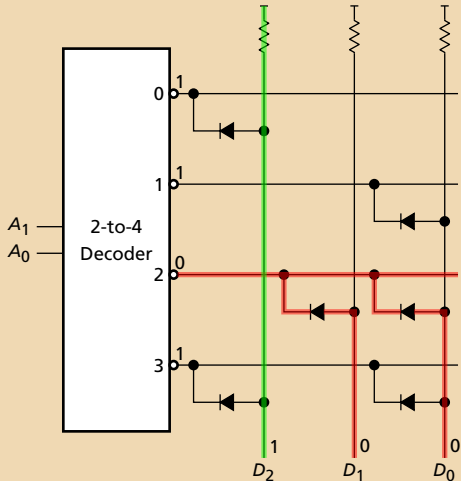


# Implementing ROMs



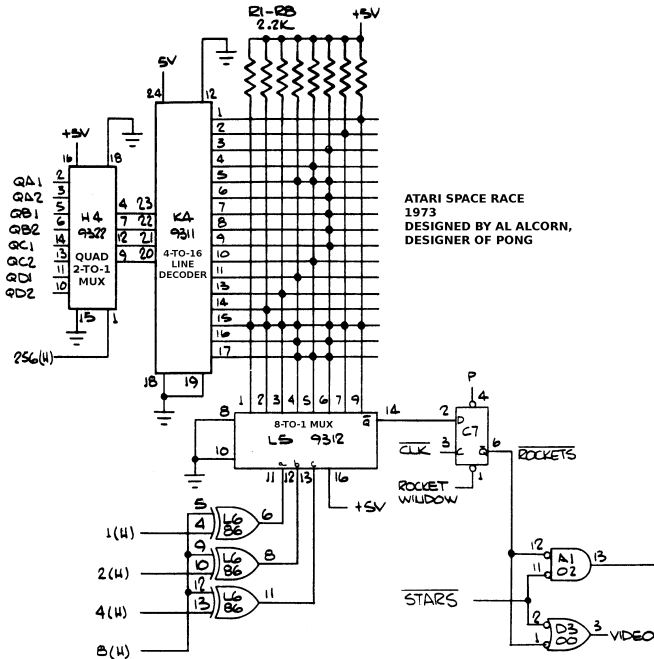
## Add. Data

00	011
01	110
10	100
11	010

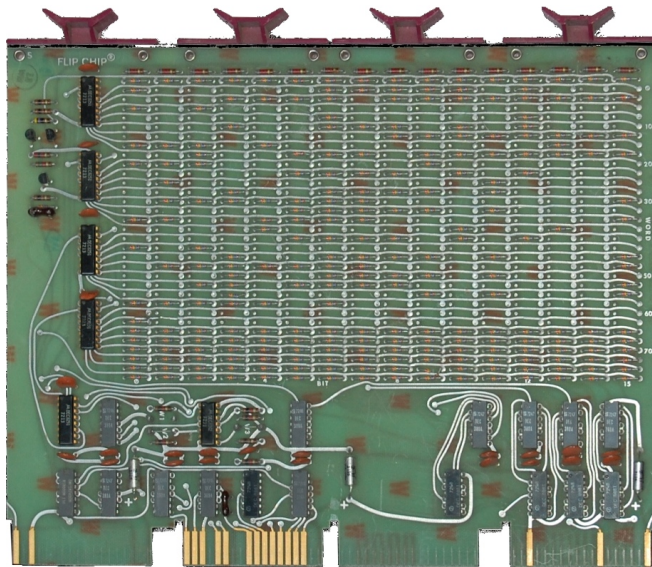




# Atari Space Race Schematic



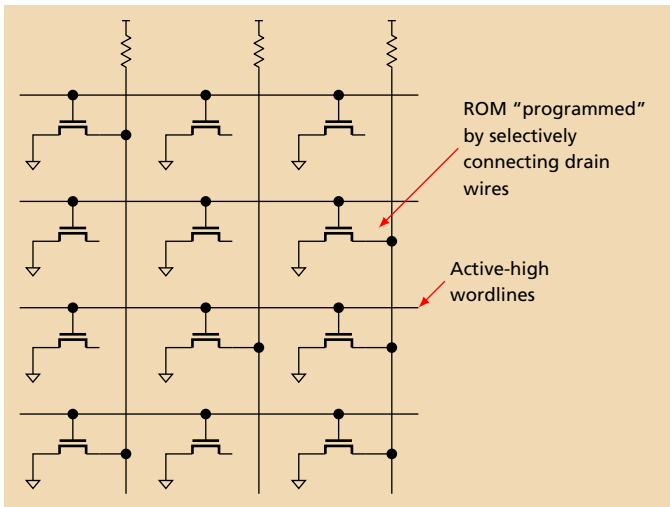
# The 1971 DEC M792-YB Bootstrap Diode Matrix



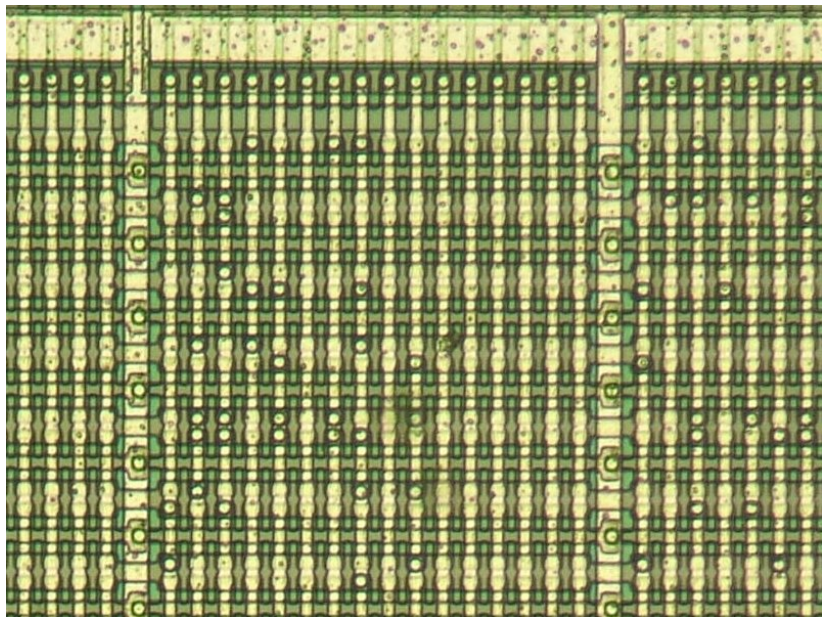
32-word, 16-bit (64-byte) ROM diode matrix

# CMOS Mask-Programmed ROMs

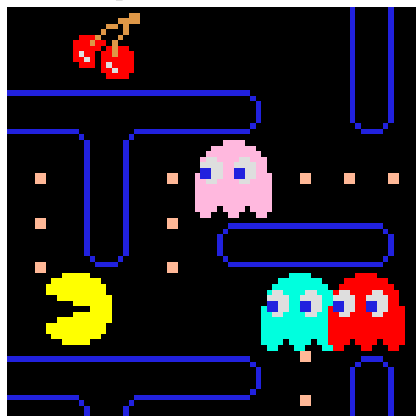
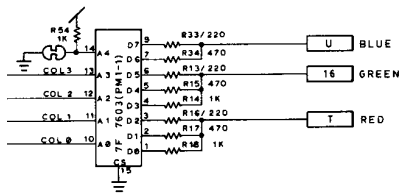
Add. Data	
00	011
01	110
10	100
11	010



# Mask ROM Die Photo

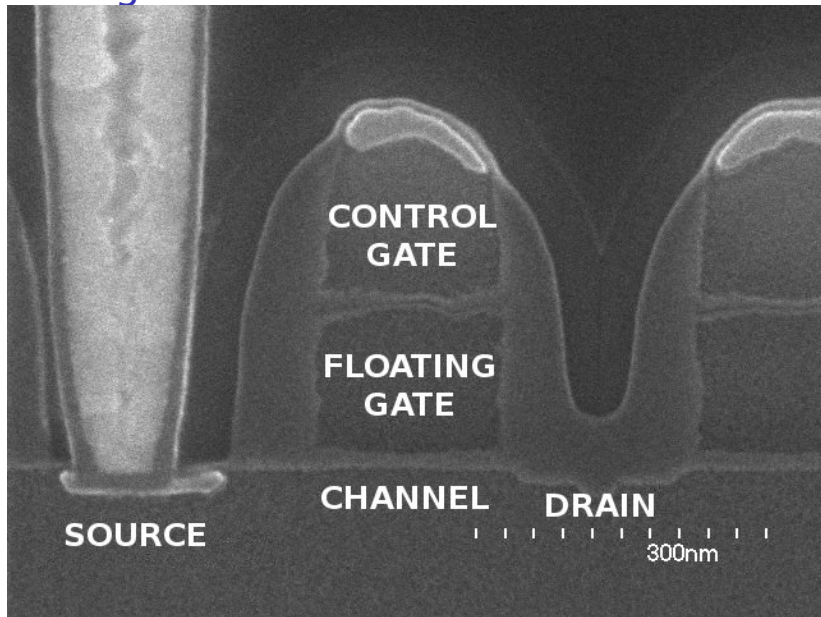


# Color PROM in Pac-Man



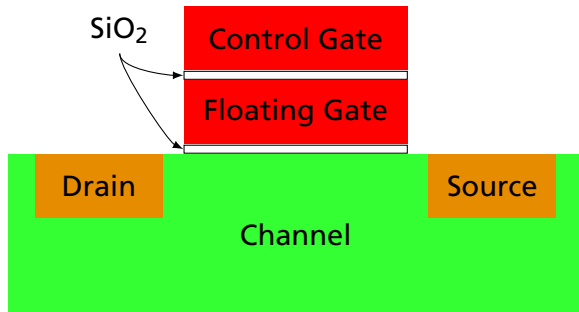
00	00	█
01	07	█
02	66	█
03	EF	█
04	00	█
05	F8	█
06	EA	█
07	6F	█
08	00	█
09	3F	█
0A	00	█
0B	C9	█
0C	38	█
0D	AA	█
0E	AF	█
0F	F6	█
10	00	█
⋮	⋮	
1F	00	█

# A Floating Gate MOSFET



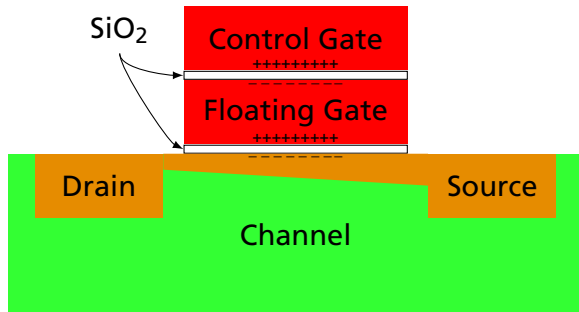
Cross section of a NOR FLASH transistor. Kawai et al., ISSCC 2008 (Renesas)

# Floating Gate n-channel MOSFET



Floating gate uncharged; Control gate at 0V: Off

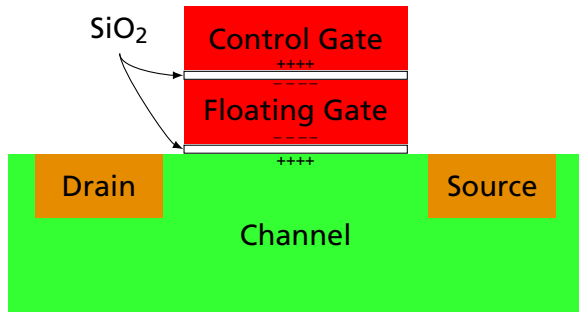
# Floating Gate n-channel MOSFET



Floating gate uncharged; Control gate positive: On

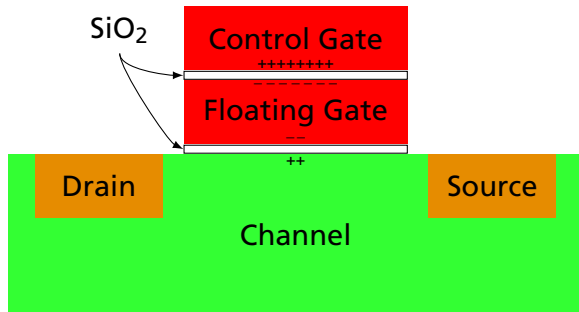


# Floating Gate n-channel MOSFET



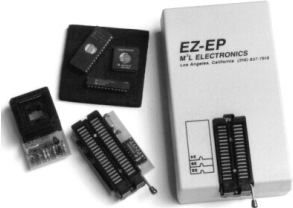
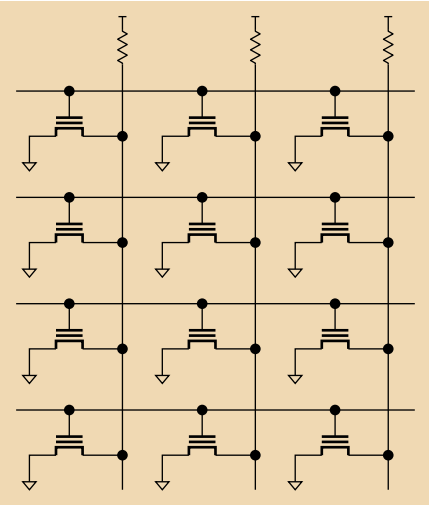
Floating gate negative; Control gate at 0V: Off

# Floating Gate n-channel MOSFET

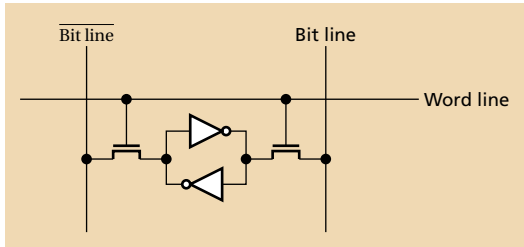


Floating gate negative; Control gate positive: Off

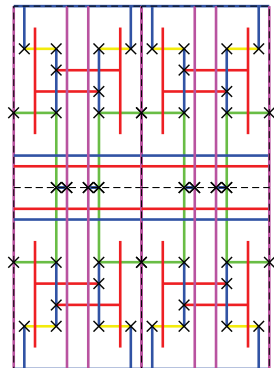
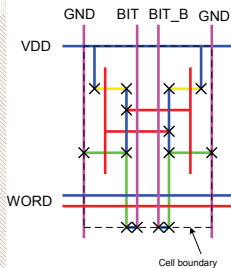
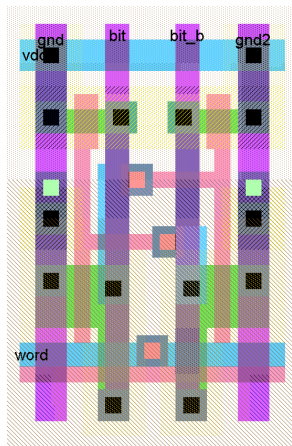
# EPROMs and FLASH use Floating-Gate MOSFETs



# Static Random-Access Memory Cell

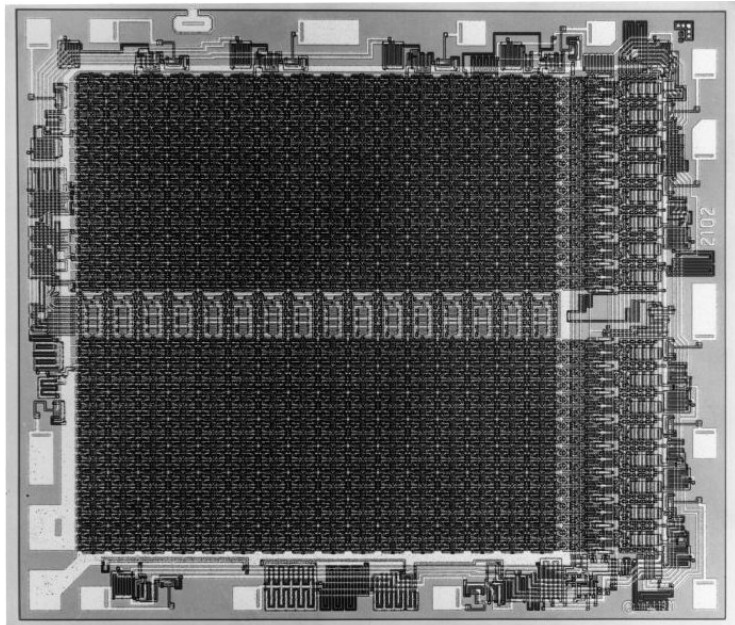


# Layout of a 6T SRAM Cell

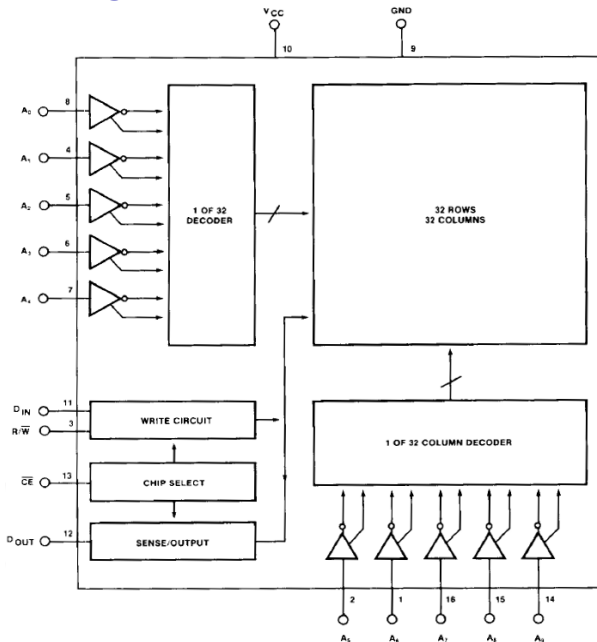


Weste and Harris. *Introduction to CMOS VLSI Design*. Addison-Wesley, 2010.

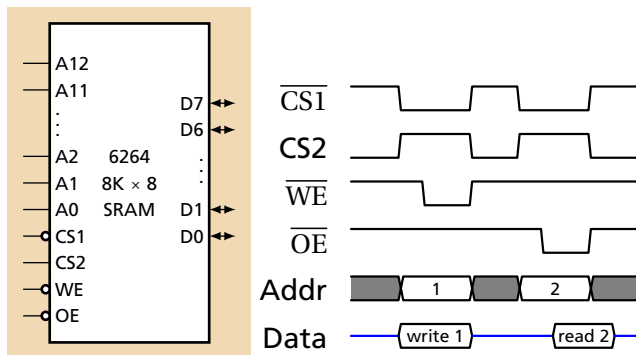
# Intel's 2102 SRAM, 1024 × 1 bit, 1972



# 2102 Block Diagram

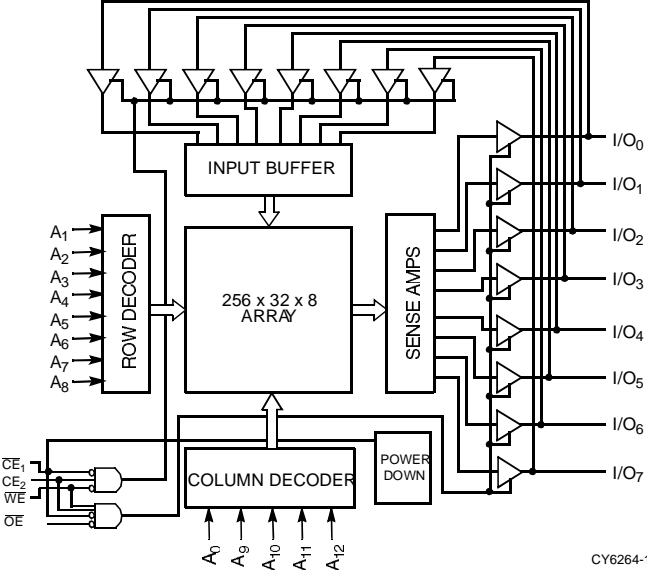


# SRAM Timing

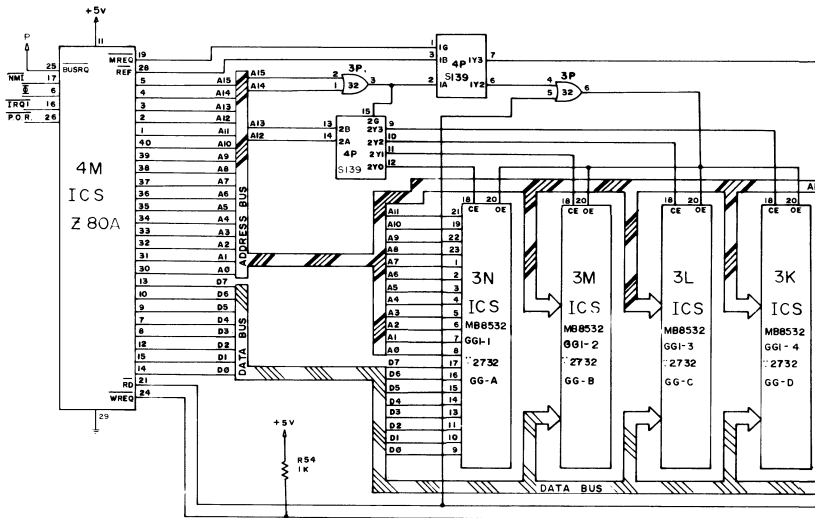




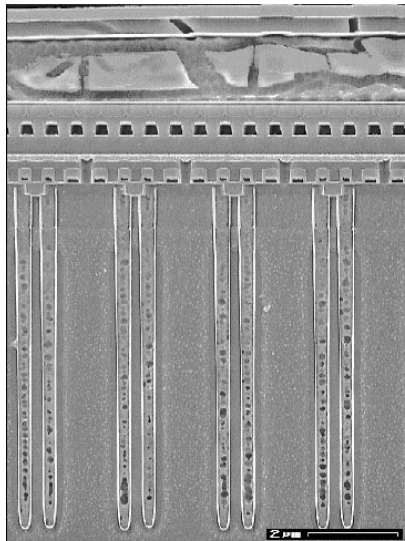
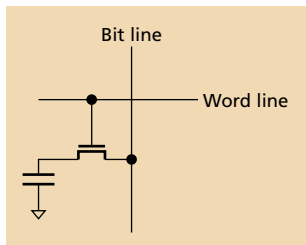
# 6264 SRAM Block Diagram



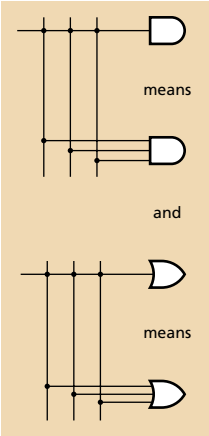
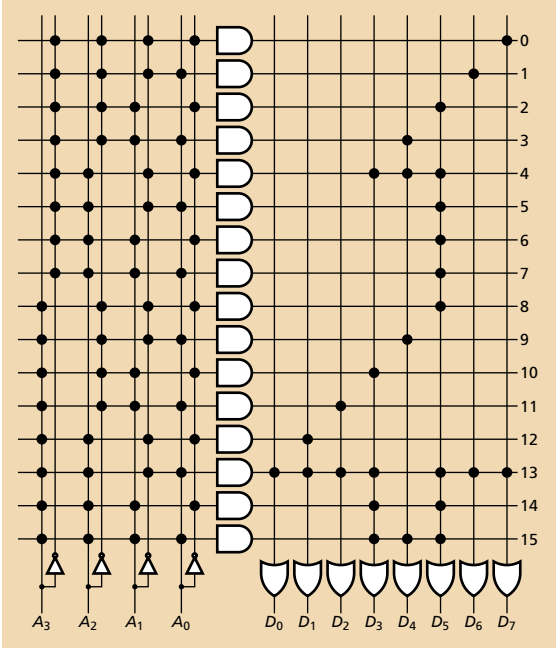
# Galaga CPU detail (Namco/Midway 1981)



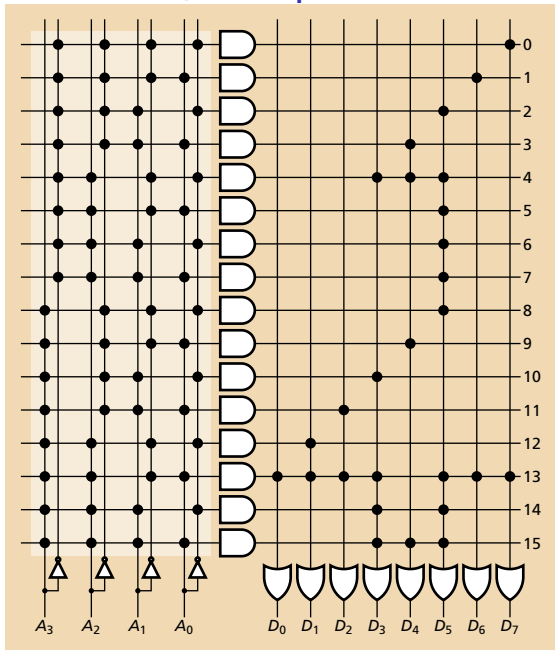
# Dynamic RAM Cell



# Our Old Pal, the Space Race ROM



# Our Old Pal, the Space Race ROM

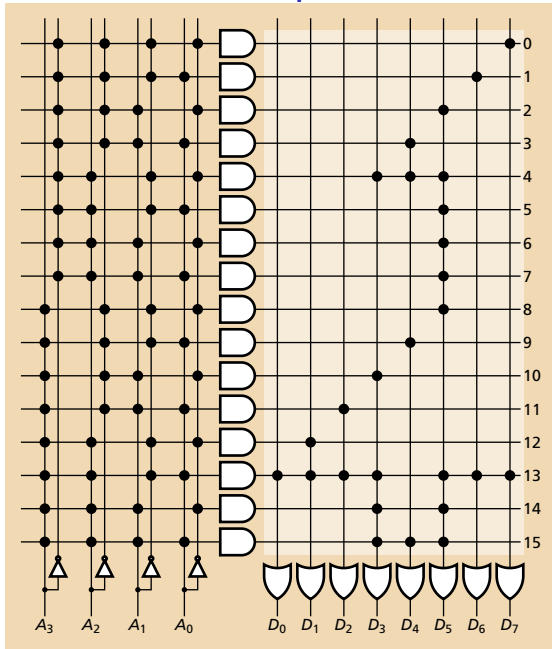


The decoder or  
"AND plane"

In a RAM or ROM,  
computes every  
minterm

Pattern is not  
programmable

# Our Old Pal, the Space Race ROM

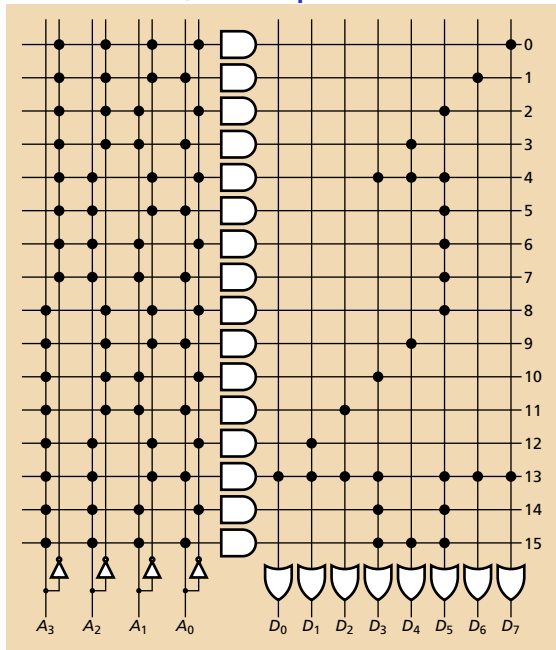


The decoder or “OR plane”

One term for every output

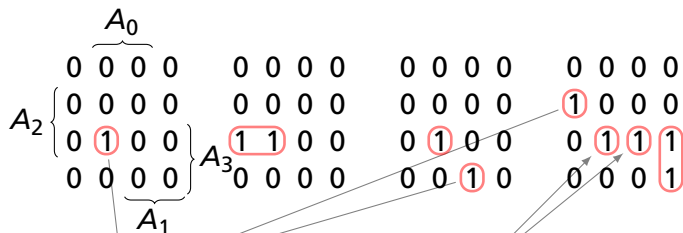
Pattern is programmable = the contents of the ROM

# Our Old Pal, the Space Race ROM



Can we do better?

# Simplifying the Space Race ROM



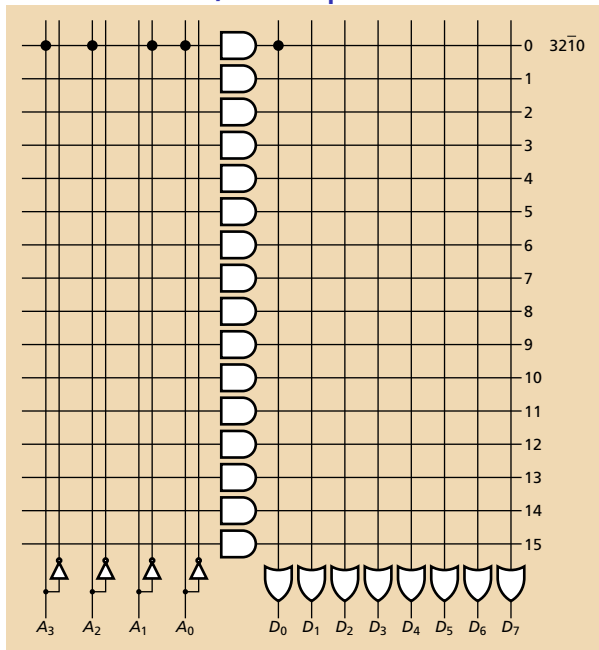
Essential minterms

mean don't expand these



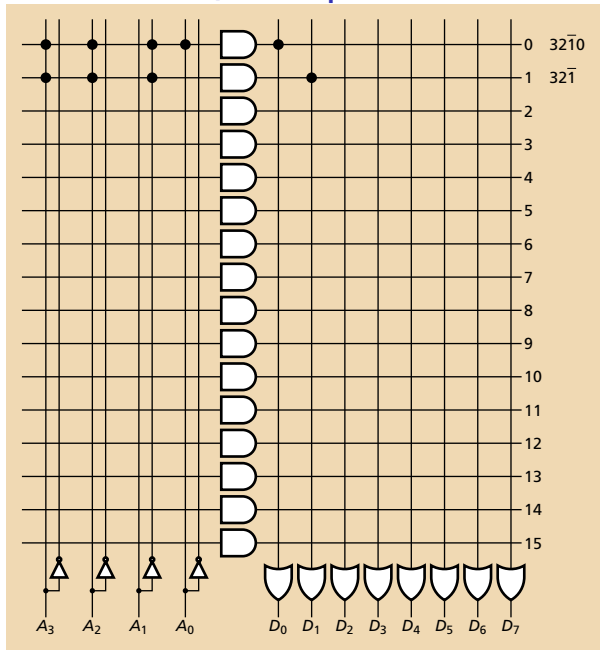


# Our New PAL, the Space Race ROM



$$D_0 = 32\bar{1}0$$

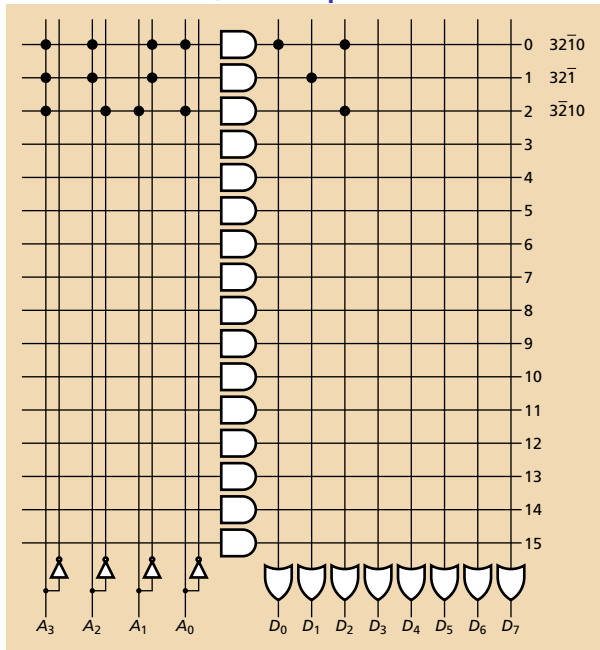
# Our New PAL, the Space Race ROM



$$D_0 = 32\bar{1}0$$

$$D_1 = 32\bar{1}$$

# Our New PAL, the Space Race ROM

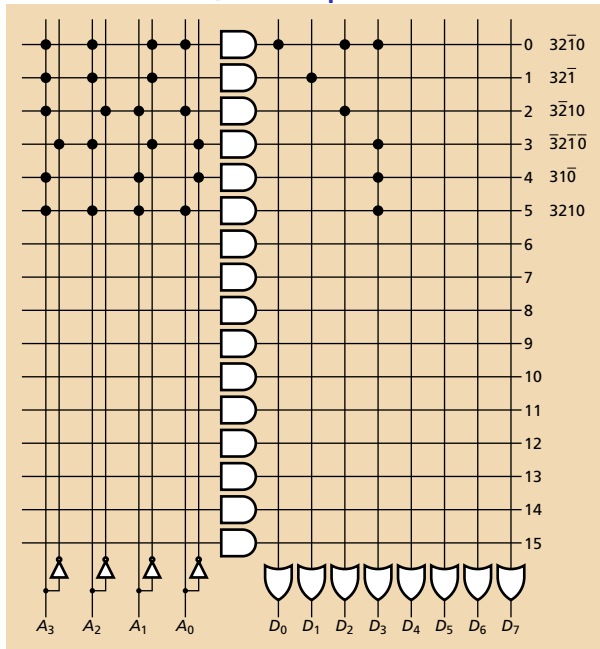


$$D_0 = 3\bar{2}10$$

$$D_1 = 3\bar{2}1$$

$$D_2 = 3\bar{2}10 + 3\bar{2}10$$

# Our New PAL, the Space Race ROM



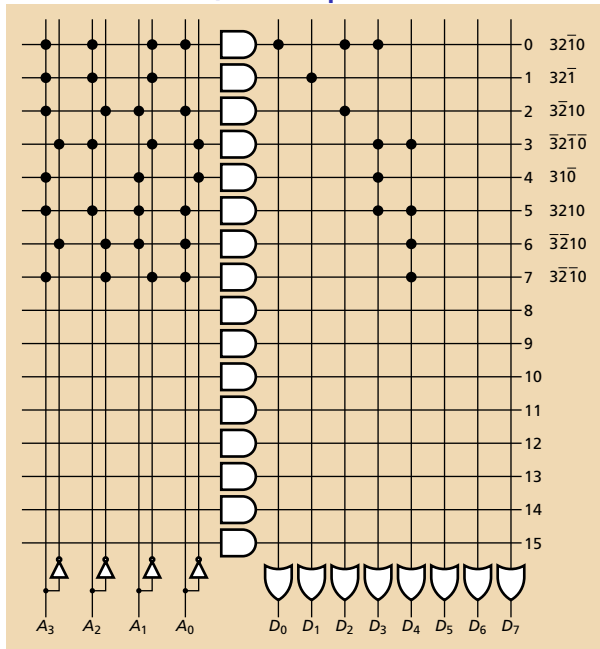
$$D_0 = 3\bar{2}\bar{1}0$$

$$D_1 = 3\bar{2}\bar{1}$$

$$D_2 = \bar{3}\bar{2}1\bar{0} + 3\bar{2}\bar{1}0$$

$$D_3 = \bar{3}\bar{2}\bar{1}\bar{0} + 3\bar{1}\bar{0} + 3\bar{2}\bar{1}0 + 32\bar{1}0$$

# Our New PAL, the Space Race ROM



$$D_0 = 32\bar{1}0$$

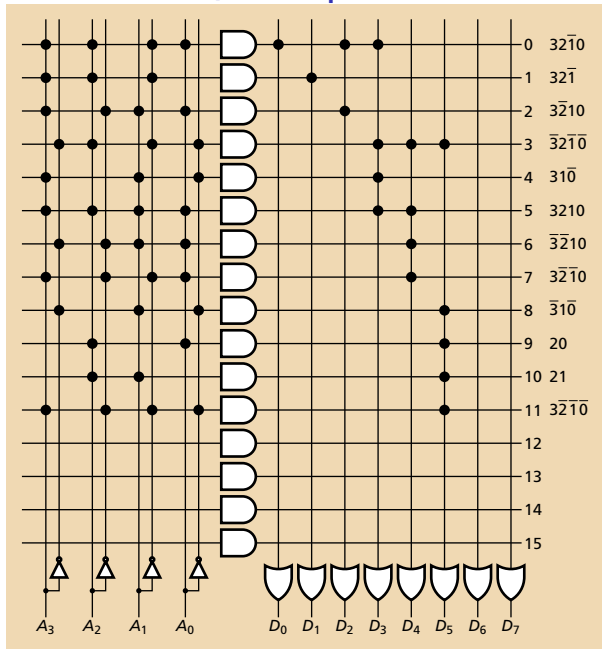
$$D_1 = 32\bar{1}$$

$$D_2 = \bar{3}2\bar{1}0 + 32\bar{1}0$$

$$D_3 = \bar{3}2\bar{1}0 + 31\bar{0} + 32\bar{1}0 + 3210$$

$$D_4 = \bar{3}2\bar{1}0 + \bar{3}2\bar{1}0 + 3\bar{2}\bar{1}0 + 3210$$

# Our New PAL, the Space Race ROM



$$D_0 = 32\bar{1}0$$

$$D_1 = 32\bar{1}$$

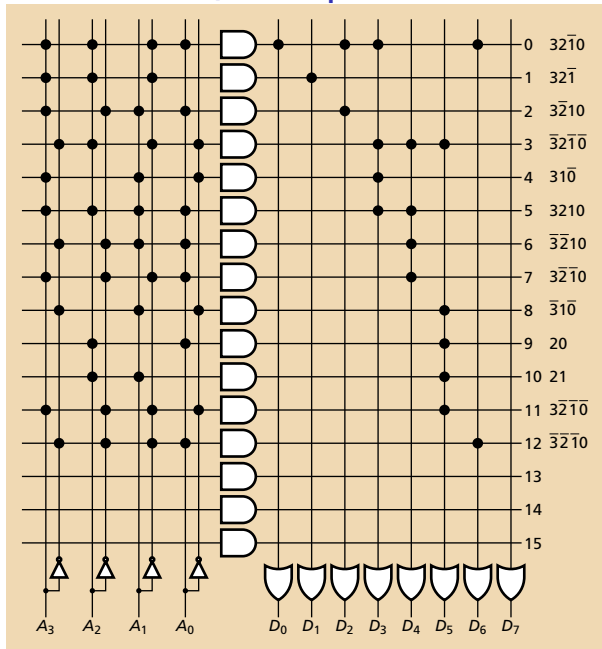
$$D_2 = 3\bar{2}10 + 32\bar{1}0$$

$$D_3 = \bar{3}2\bar{1}0 + 31\bar{0} + 3\bar{2}10 + 3210$$

$$D_4 = \bar{3}\bar{2}10 + \bar{3}2\bar{1}0 + \bar{3}2\bar{1}0 + 3210$$

$$D_5 = \bar{3}1\bar{0} + 20 + 21 + \bar{3}2\bar{1}0 + 3\bar{2}10$$

# Our New PAL, the Space Race ROM



$$D_0 = 32\bar{1}0$$

$$D_1 = 32\bar{1}$$

$$D_2 = 3\bar{2}10 + 32\bar{1}0$$

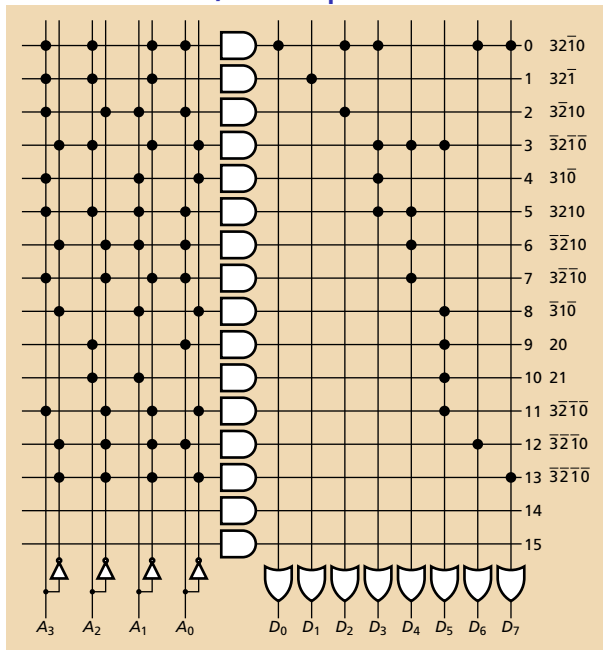
$$D_3 = \bar{3}2\bar{1}0 + 31\bar{0} + 32\bar{1}0 + 3210$$

$$D_4 = \bar{3}2\bar{1}0 + \bar{3}2\bar{1}0 + 3\bar{2}\bar{1}0 + 3210$$

$$D_5 = \bar{3}1\bar{0} + 20 + 21 + 3\bar{2}\bar{1}0 + 32\bar{1}0$$

$$D_6 = \bar{3}2\bar{1}0 + 32\bar{1}0$$

# Our New PAL, the Space Race ROM



$$D_0 = 32\bar{1}0$$

$$D_1 = 32\bar{1}$$

$$D_2 = 3\bar{2}\bar{1}0 + 32\bar{1}0$$

$$D_3 = \bar{3}\bar{2}\bar{1}0 + 31\bar{0} + 32\bar{1}0 + 3210$$

$$D_4 = \bar{3}\bar{2}\bar{1}0 + \bar{3}\bar{2}\bar{1}0 + 3\bar{2}\bar{1}0 + 3210$$

$$D_5 = \bar{3}\bar{1}0 + 20 + 21 + \bar{3}\bar{2}\bar{1}0 + \bar{3}\bar{2}\bar{1}0$$

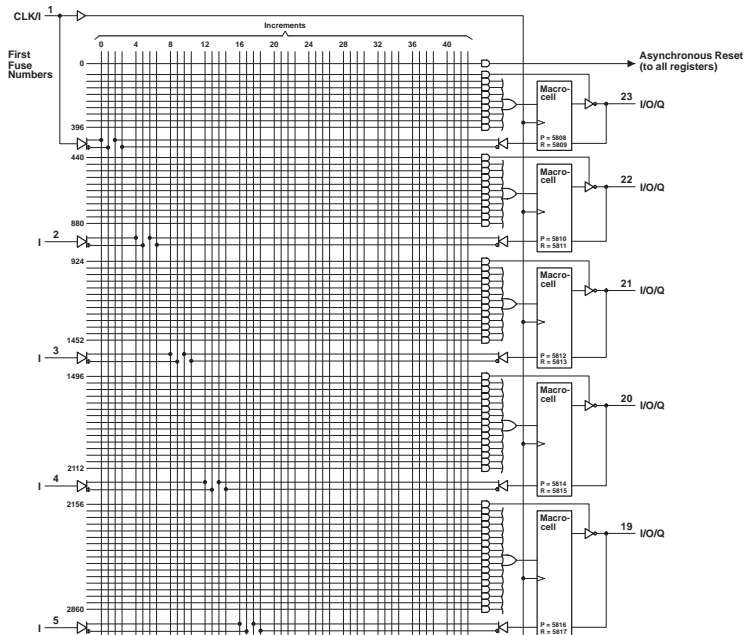
$$D_6 = \bar{3}\bar{2}\bar{1}0 + 32\bar{1}0$$

$$D_7 = \bar{3}\bar{2}\bar{1}0 + 32\bar{1}0$$

Saved two ANDs



# A 22V10 PAL: Programmable AND/Fixed OR



# Field-Programmable Gate Arrays (FPGAs)

