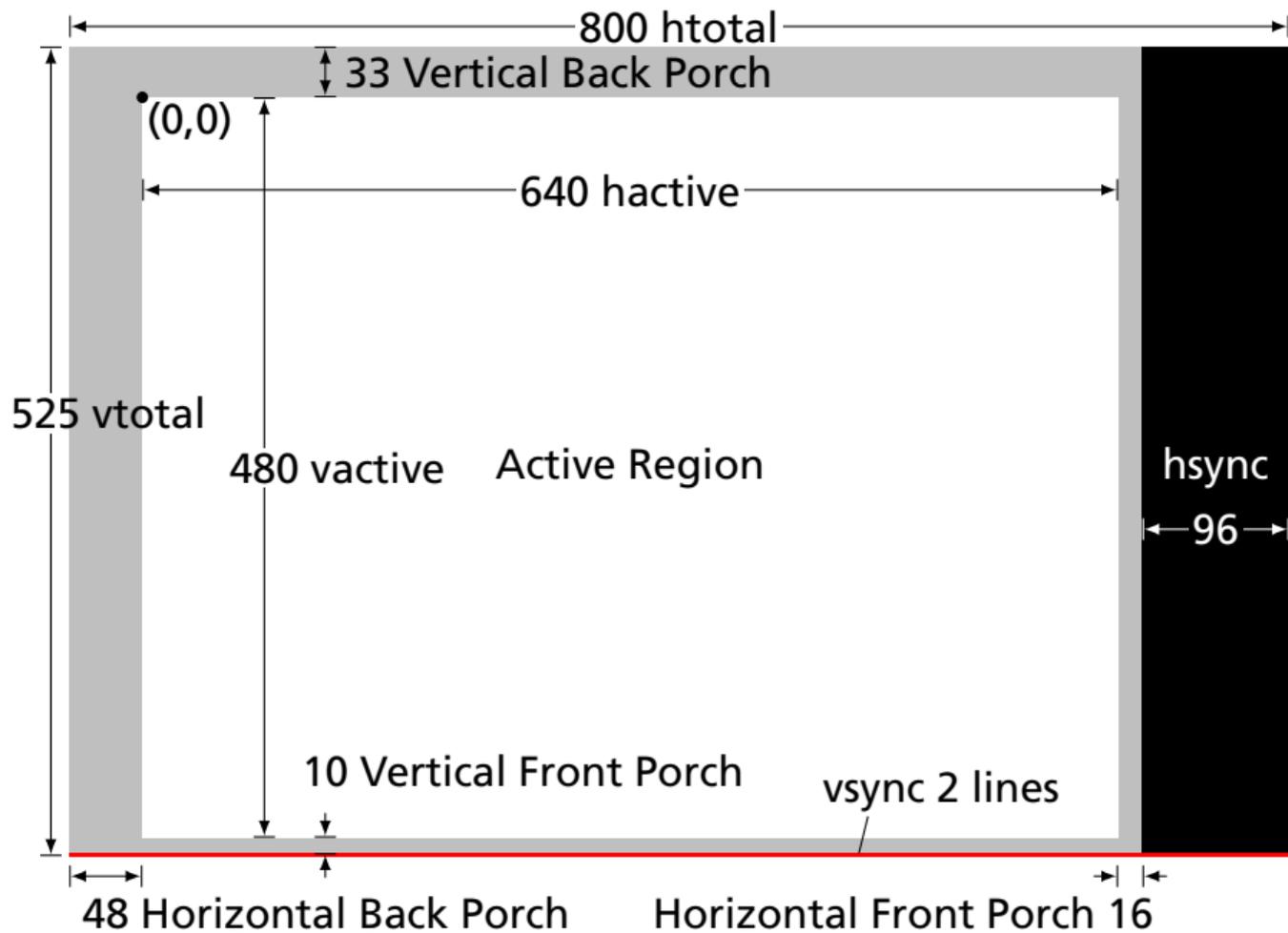


Tiles and Sprites: 1980s Videogame Graphics

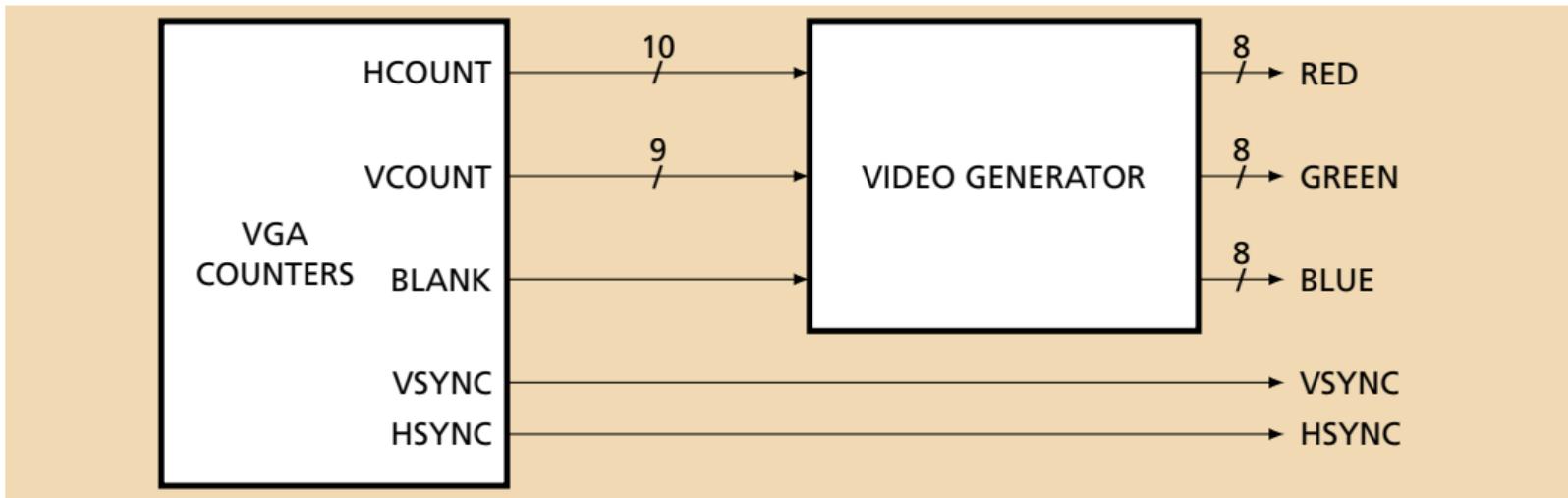
Stephen A. Edwards

Columbia University

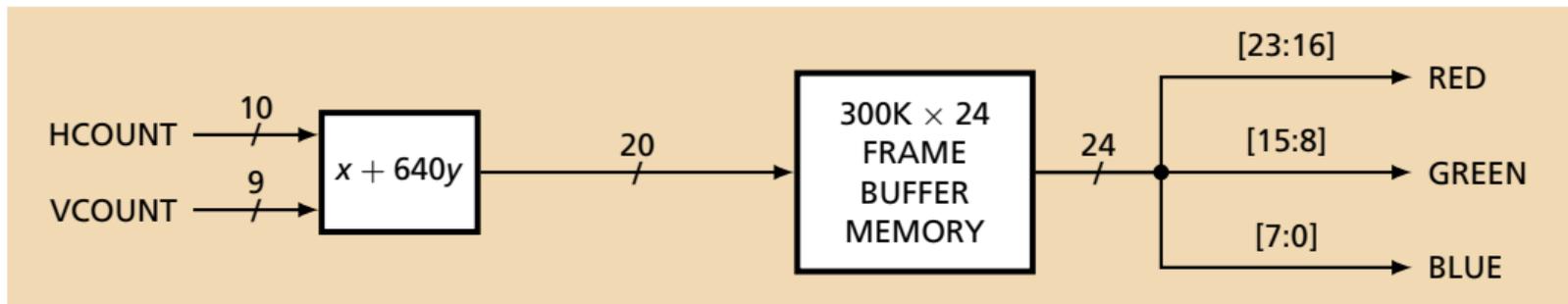
Spring 2026



VGA Video Generator: 640 × 480

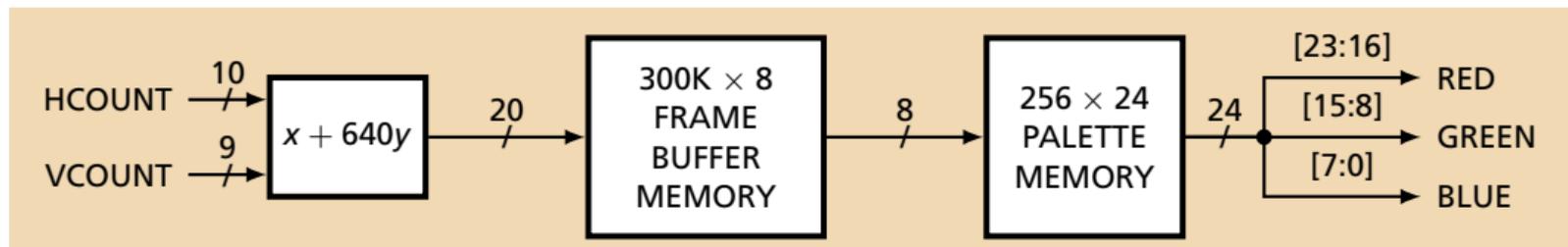


A 24 bpp "true color" VGA framebuffer



$$640 \times 480 \times 3 = 921\,600 = 900 \text{ KB}$$

An 8 bpp VGA frame buffer using indexed color



Idea: only display 256 different colors from a much larger palette

$$640 \times 480 = 307\,200 = 300 \text{ KB}$$



Pac-Man

Namco, 1980

Z80A @ 3.072 MHz

224 × 288 pixels

16 colors

16 KB program ROM

8 KB graphic ROM

3 KB program RAM

256 B scanline RAM



Excite Bike on the
Famicom/NES

Nintendo, 1984

6502 @ 1.79 MHz

256 × 240 pixels

64 colors

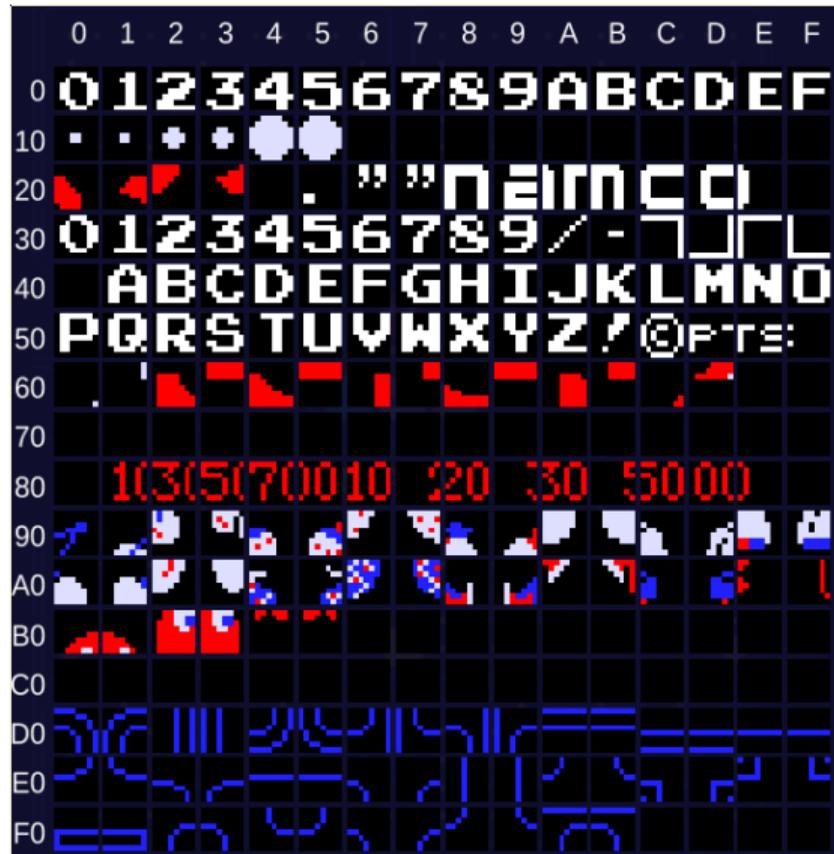
2 KB work RAM

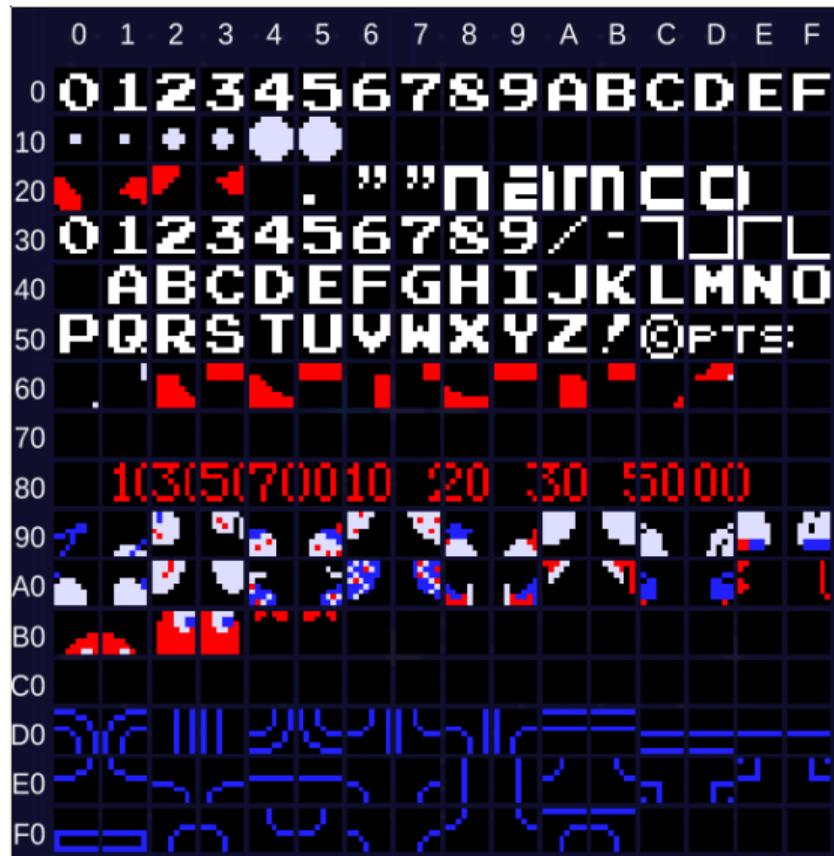
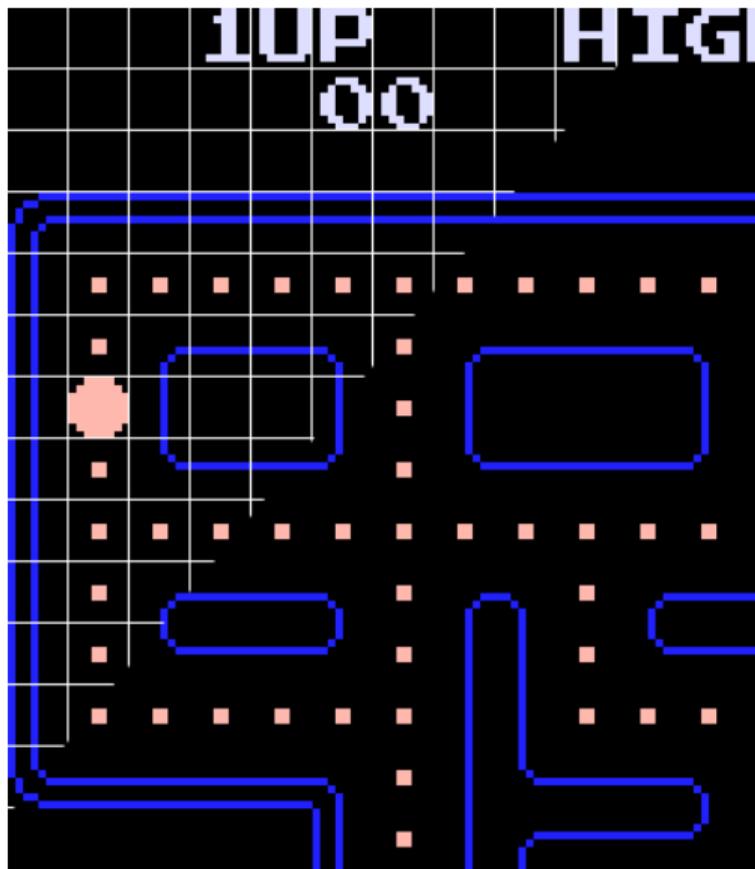
2 KB video RAM

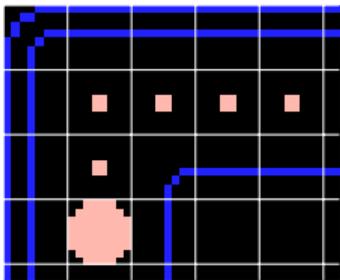
256 byte sprite RAM

16 KB ROM (cartridge)

8 KB graphics ROM
(cartridge)







Image

		c				
		0	1	2	3	4
r	0	D1	DA	DA	DA	DA
	1	D3	10	10	10	10
	2	D3	10	E7	DE	DE
	3	D3	14	E9	FC	FC

Tile Map



Tile Set

		x									
		0	1	2	3	4	5	6	7	8	9
y	0	(0,0)	(1,0)						(7,0)	(0,0)	(1,0)
	1	(0,1)									
	2										
	3										
	4										
	5										
	6										
	7	(0,7)							(7,7)	(0,7)	
	8	(0,0)							(7,0)	(0,0)	(1,0)
	9	(0,1)								(0,1)	

Pixel Coordinates

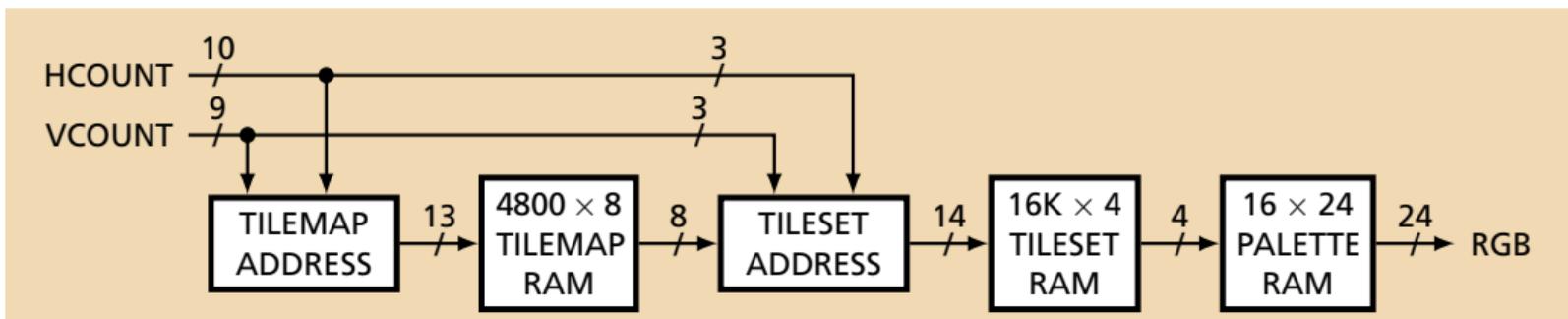
Tile Generator Design

8×8 tiles

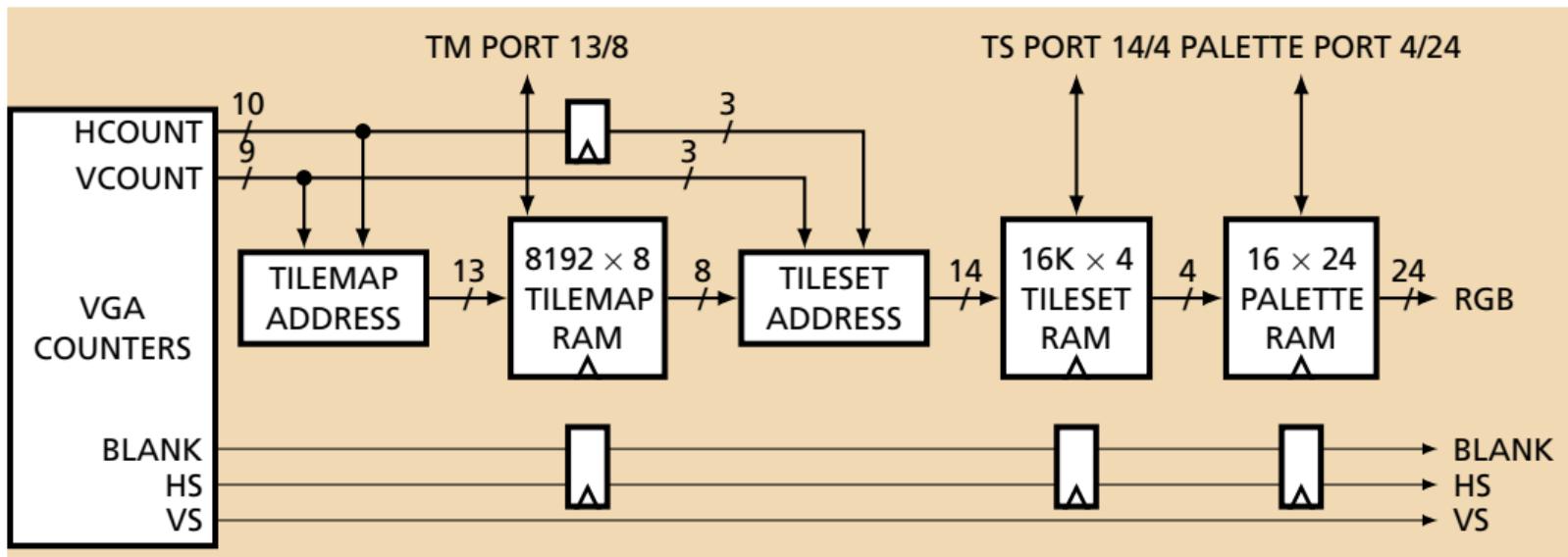
$$640 \div 8 = 80 \quad 480 \div 8 = 60$$

$$80 \times 60 = 4800 \approx 5K$$

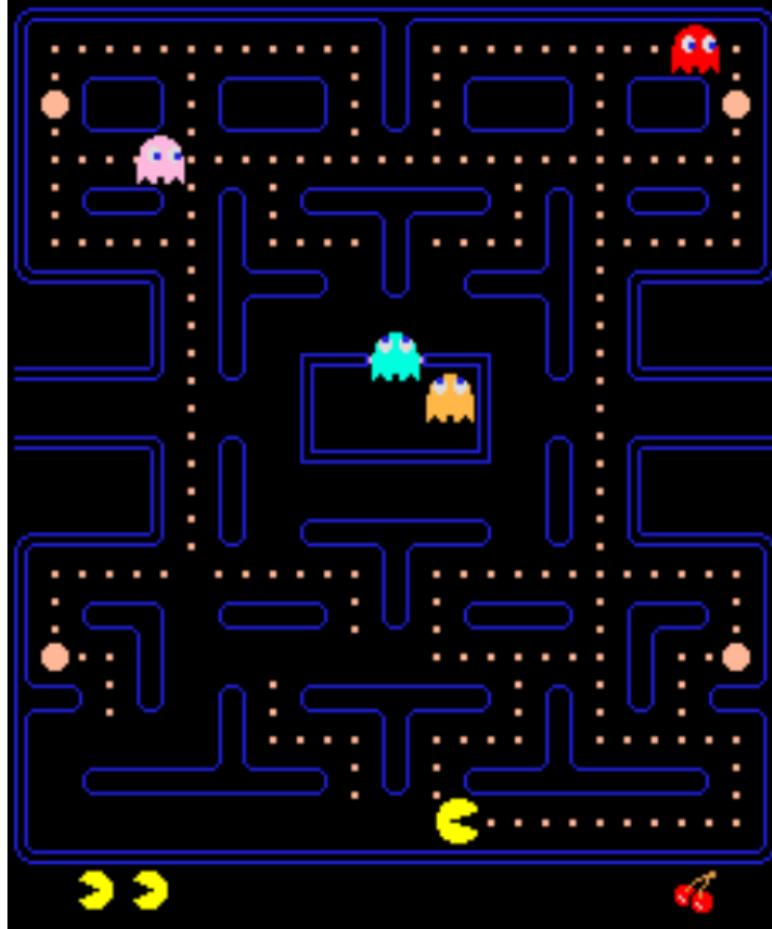
$$4 \text{ bpp} \times 256 \text{ tiles} \times 8 \times 8 = 64 \text{ kb} = 8 \text{ KB}$$

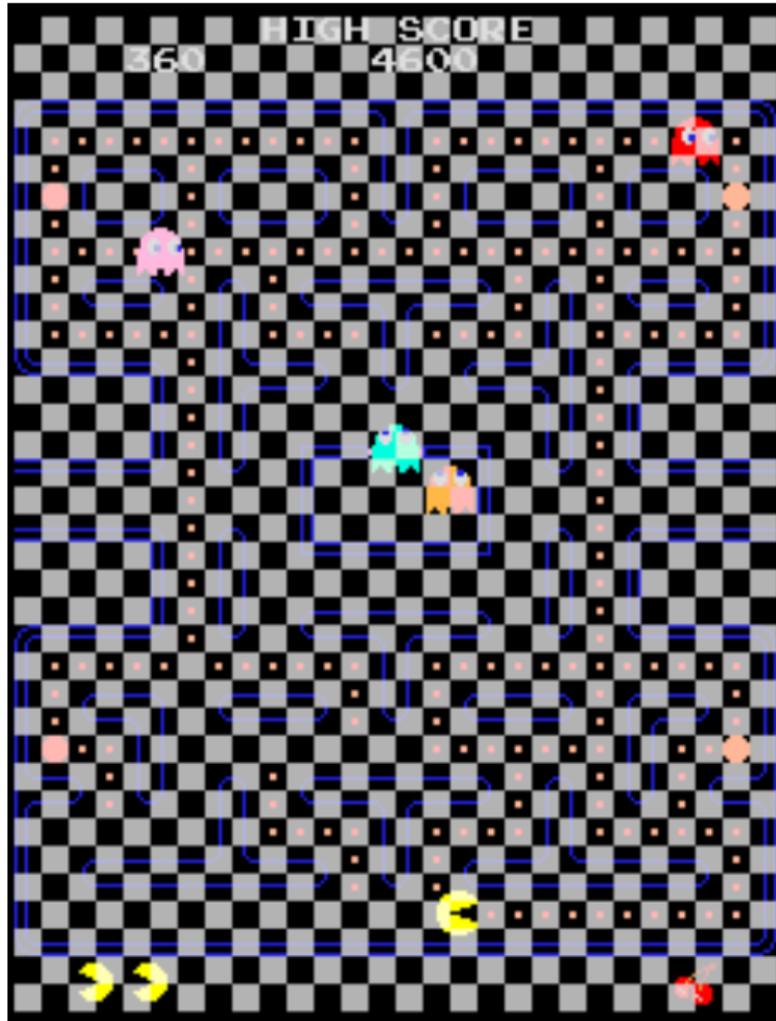


Pipelining the Tile Generator; Adding Access Ports



360 HIGH SCORE 4600

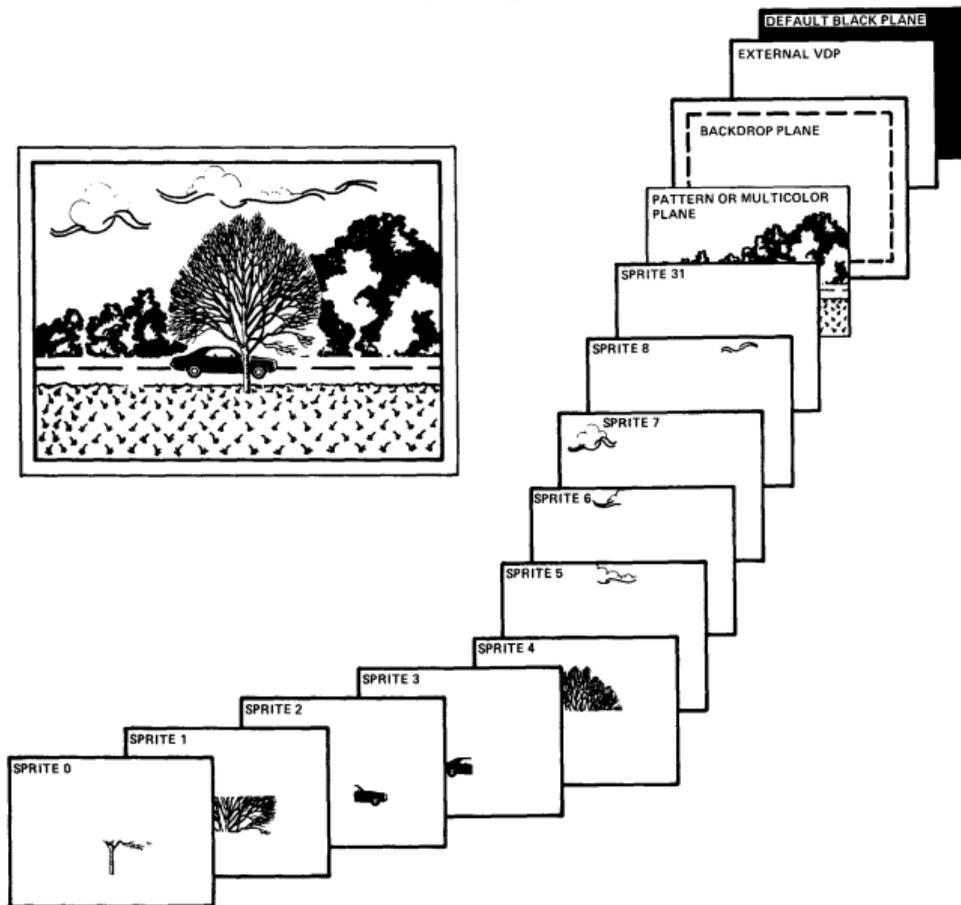




The ghosts spoil everything

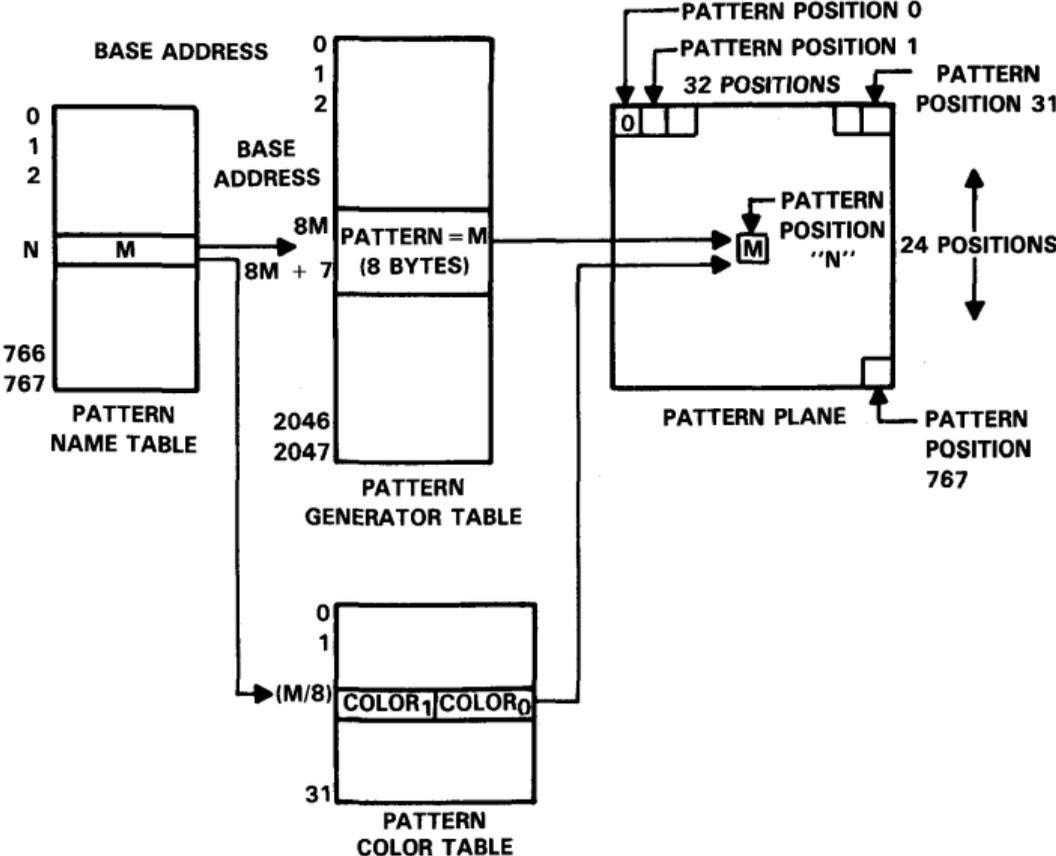
They need to move smoothly: need pixel-level accuracy

TMS9918 Video Display Processor

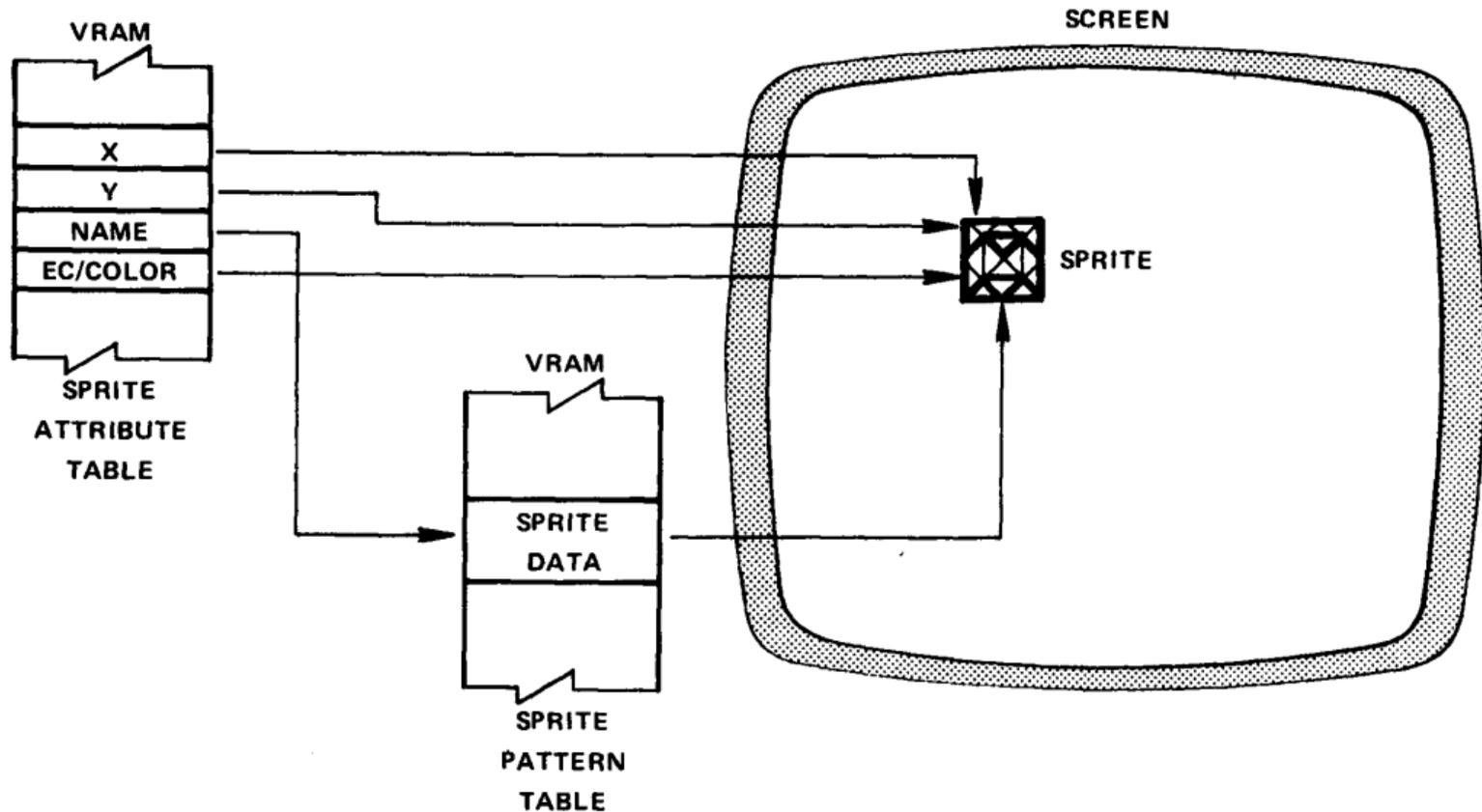


TI Video Display Processor
Programmer's Guide, 1984

TMS9918 Tiles

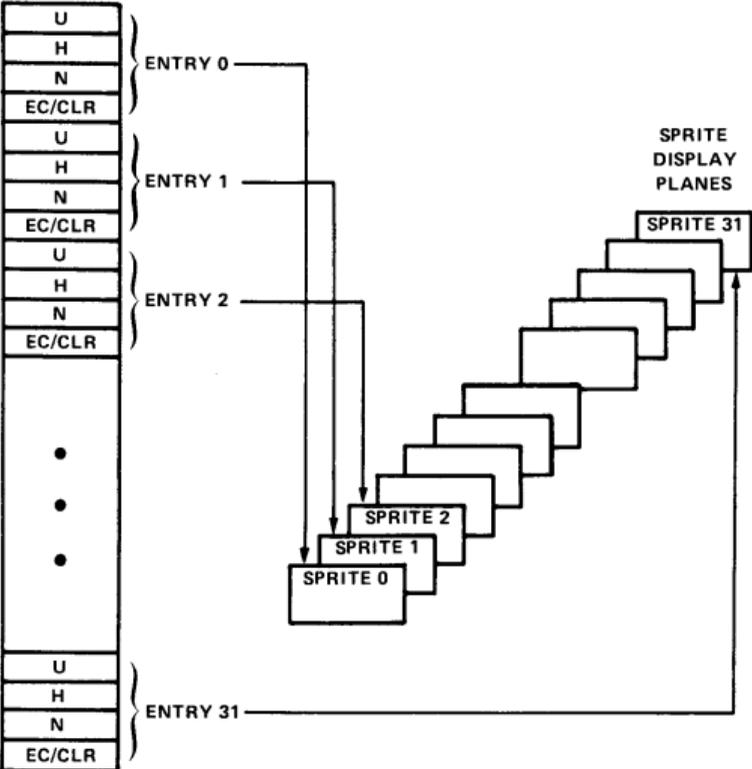


TMS9918 Sprite Generation

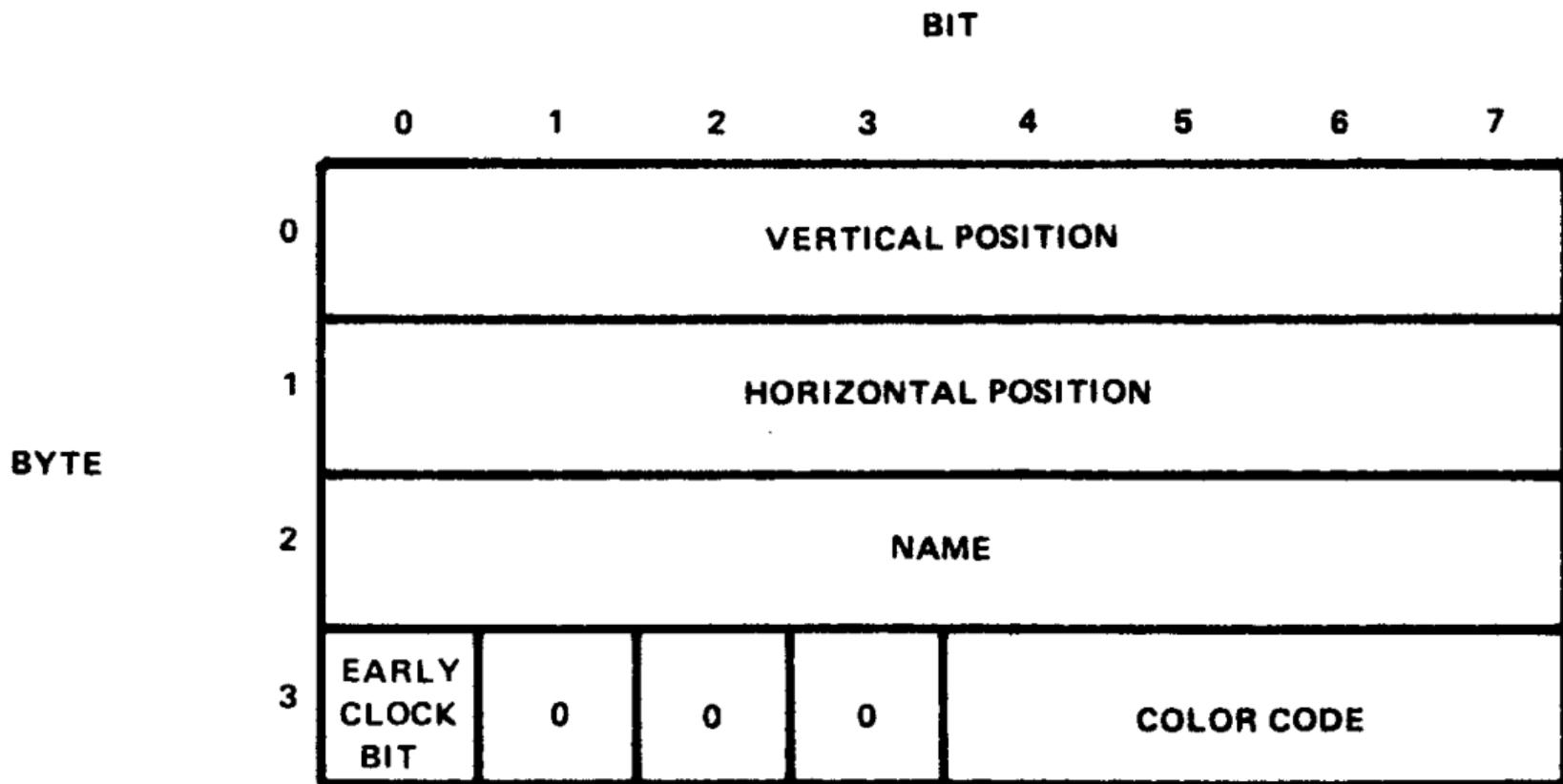


TMS9918 Sprite Attribute Table

SPRITE
ATTRIBUTE
TABLE



TMS9918 Sprite Attribute Table Entry



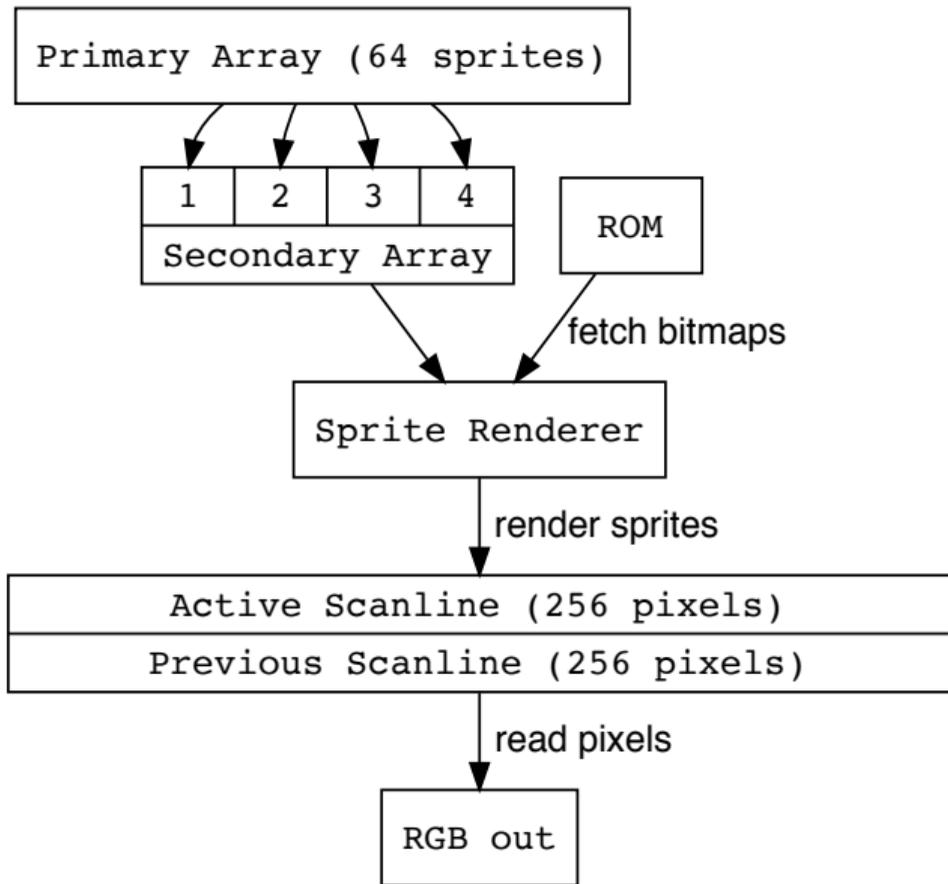


Figure 32.1: Data flow of sprite scanline renderer.

Steven Hugg, Designing Video Game Hardware in Verilog, 8bitworkshop, 2018.