How to Program in Assembly Language

for the

COMMANDER X16™

Lesson 12: Tiles

VERA Tile Modes

- Each VERA layer can be any tile or bitmap mode
- Variable map size
- Text: 1 bit-per-pixel (1bpp), 8x8 pixels, Max = 256 characters
  - 16-color: background, foreground
  - 256-color: foreground; background = black
- Graphics: Variable bpp, variable tile size, Max = 1024 characters
  - 2bpp, 4-color graphics
  - 4bpp, 16-color graphics
  - 8bpp, 256-color graphics

Tile Layer Configuration

- **L0_CONFIG**: $9F2D**; L1_CONFIG**: $9F34
Map Height	Map Width	T256	Bitmap Mode	Color Depth
    - Map Height/Width: 0 = 32 tiles; 1 = 64 tiles; 2 = 128 tiles; 3 = 256 tiles
    - Color Depth: 0 = 1bpp; 1 = 2bpp; 2 = 4bpp; 3 = 8bpp
    - T256: Color Depth == 1bpp → 0 = 16-color; 1 = 256-color
    - Bitmap Mode: 0 = text/tile mode; 1 = bitmap mode
- **L0_MAPBASE**: $9F2E; **L1_MAPBASE**: $9F35
  - Tile map start address >> 9 (512-byte alignment)
- **L0_TILEBASE**: $9F2F; **L1_TILEBASE**: $9F36
Tile Base Address (16:11)	Tile Height	Tile Width
    - Tile graphics base address >> 11 (2kB alignment)
    - Tile Height/Width: 0 = 8 pixels; 1 = 16 pixels

2bpp Tiles

- Each byte: 4 pixels
  - 8x8 tile: 16 bytes
  - 16x16 tile: 64 bytes
- Colors: First four of any 16-color palette offset
  - Default: 0 1 2 3
  - 0-3 remapped to 16-19, 32-35, 48-51, etc.
- Good enough for…
  - CGA!
  - NES!

00	00	00	10	10	00	00	00
00	00	10	10	10	10	00	00
00	10	10	11	11	10	10	00
10	10	11	01	01	11	10	10
10	10	11	01	01	11	10	10
00	10	10	11	11	10	10	00
00	00	10	10	10	10	00	00
00	00	00	10	10	00	00	00

02 80
0A A0
2B E8
AD 7A
AD 7A
2B E8
0A A0
02 80

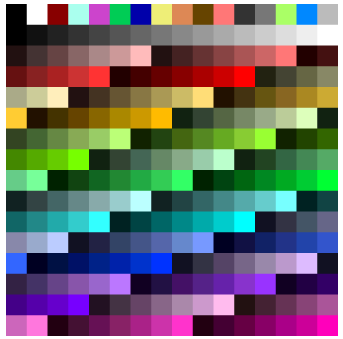
4bpp Tiles

- Each byte: 2 pixels (nybble = pixel)
  - 8x8 tile: 32 bytes
  - 16x16 tile: 128 bytes
- Colors: Any 16-color palette offset
  - Default: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  - 0-15 remapped to 16-31, 32-47, 48-63, etc.
- Good enough for…
  - EGA!
  - SNES (Modes 1, 2, 3, 5, 6)!

0	0	0	2	2	0	0	0
0	0	2	8	8	2	0	0
0	2	8	7	7	8	2	0
2	8	7	5	5	7	8	2
8	7	5	6	6	5	7	8
7	5	6	4	4	6	5	7
5	6	4	0	0	4	6	5
6	4	0	0	0	0	4	6

00 02 20 00
00 28 82 00
02 87 78 20
28 75 57 82
87 56 65 78
75 64 46 57
56 40 04 65
64 00 00 46

8bpp Tiles

- Each byte: 1 pixel
  - 8x8 tile: 64 bytes
  - 16x16 tile: 256 bytes
- Colors: Full 256-color palette
  - Default: 
- Good enough for…
  - VGA!
  - SNES (Modes 3, 4, 7)!

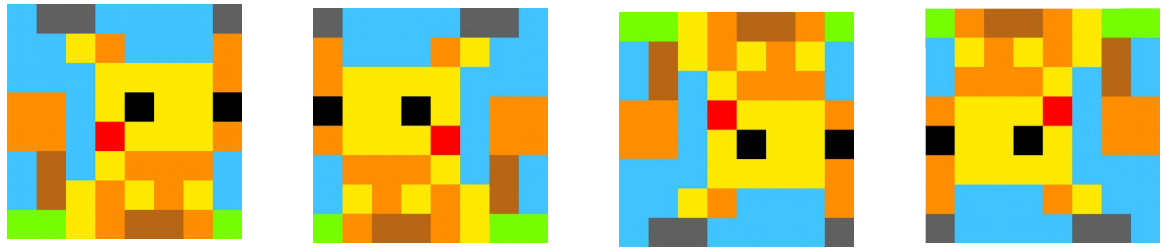
9C	0C	0C	9C	9C	9C	9C	0C
9C	9C	50	57	9C	9C	9C	57
9C	9C	9C	50	50	50	50	57
57	57	9C	50	10	50	50	10
57	57	9C	3B	50	50	50	57
9C	08	9C	50	57	57	57	9C
9C	08	50	57	50	57	50	9C
73	73	50	57	08	08	57	73

9C 0C 0C 9C 9C 9C 9C 0C
9C 9C 50 57 9C 9C 9C 57
9C 9C 9C 50 50 50 50 57
57 57 9C 50 10 50 50 10
57 57 9C 3B 50 50 50 57
9C 08 9C 50 57 57 57 9C
9C 08 50 57 50 57 50 9C
73 73 50 57 08 08 57 73

Graphical Tile Map Entry

Byte 0	Tile Index (7:0)			
Byte 1	Palette Offset	V-Flip	H-Flip	Tile Index (9:8)

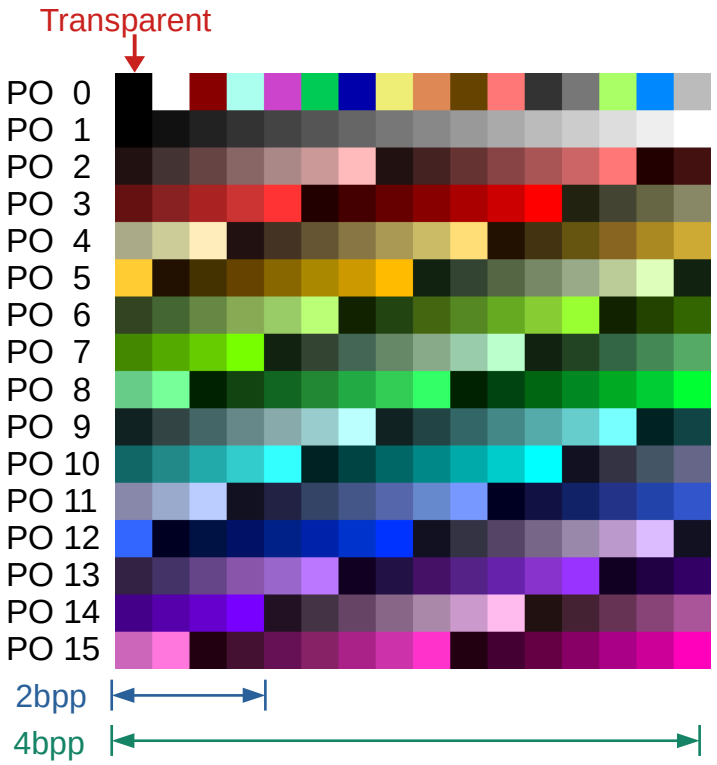
- 10-bit Tile Index (0 – 1023)
  - Tile address = TILEBASE + Tile_Index * Tile_Size
- Palette Offset: 0-15
  - Offset = Starting_Index / 16
- Flipping:



V:0, H:0 V:0, H:1 V:1, H:0 V:1, H:1

Palette Offsets

- 2bpp and 4bpp tiles and bitmaps require specific palette offset (0-15)
- Color index 0 transparent regardless of offset
- Each tile can have its own offset
  - 2bpp tile map: up to 48 unique colors + transparency
  - 4bpp tile map: up to 240 unique colors + transparency



0	0	0	2	2	0	0	0
0	0	2	8	8	2	0	0
0	2	8	7	7	8	2	0
2	8	7	5	5	7	8	2
8	7	5	6	6	5	7	8
7	5	6	4	4	6	5	7
5	6	4	0	0	4	6	5
6	4	0	0	0	0	4	6

PO 0

0	0	0	2	2	0	0	0
0	0	2	8	8	2	0	0
0	2	8	7	7	8	2	0
2	8	7	5	5	7	8	2
8	7	5	6	6	5	7	8
7	5	6	4	4	6	5	7
5	6	4	0	0	4	6	5
6	4	0	0	0	0	4	6

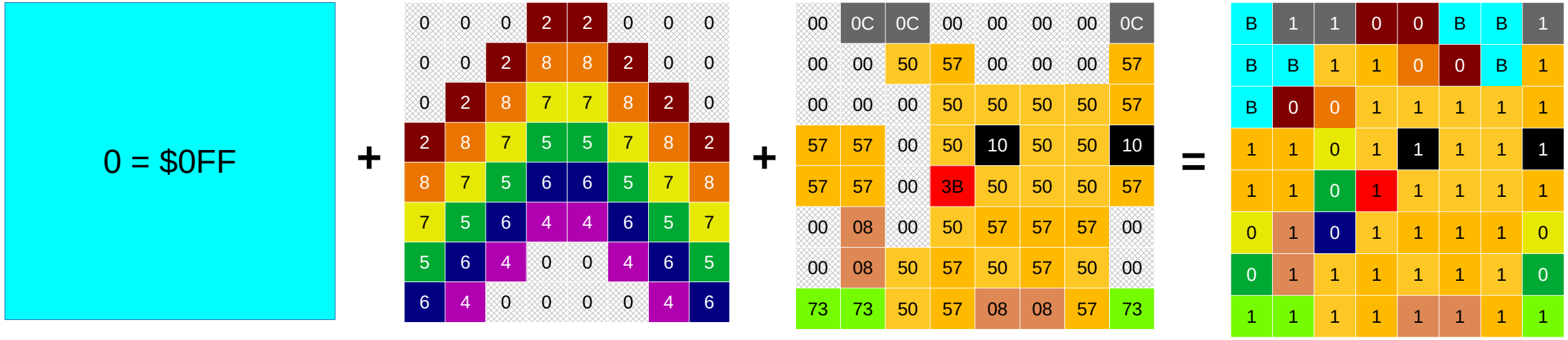
PO 1

0	0	0	2	2	0	0	0
0	0	2	8	8	2	0	0
0	2	8	7	7	8	2	0
2	8	7	5	5	7	8	2
8	7	5	6	6	5	7	8
7	5	6	4	4	6	5	7
5	6	4	0	0	4	6	5
6	4	0	0	0	0	4	6

PO 15

Transparency

- Palette color 0 (at $1FA00, default = black) is background color.
- Tiles (and bitmaps!) using color index 0 will be transparent at those pixels
  - Regardless of offset: colors at indexes 16, 32, etc. ignored except for 8bpp assets
- If all visible tiles, bitmaps and sprites have 0 as color index for a pixel, background color will show through.
- Example: Change background color to cyan ($0FF) and use 4bpp & 8bpp layers



Tile Layer Scrolling

- **L0_HSCROLL_L**: $9F30**; L1_HSCROLL_L**: $9F37
  - H-Scroll (7:0)
- **L0_HSCROLL_H**: $9F31**; L1_HSCROLL_H**: $9F38
  -

Unused	H-Scroll (11:8)

- **L0_VSCROLL_L**: $9F32**; L1_VSCROLL_L**: $9F39
  - V-Scroll (7:0)
- **L0_VSCROLL_H**: $9F33**; L1_VSCROLL_H**: $9F3A
  -

Unused	V-Scroll (11:8)

- H-Scroll & V-Scroll default = 0 for both layers
- H-Scroll: Pixel column of tile map to display at left end of screen
  - H-Scroll % Tile_Width > 0 → Leftmost tiles partially rendered
- V-Scroll: Pixel row of tile map to display at top of screen
  - V-Scroll % Tile_Height > 0 → Topmost tiles partially rendered

Example Program

Parallax Scrolling

Sky 30 pixels/sec

Ground 60 pixels/sec

Display Scale: 4X (160x120)

Layers (both 0 and 1):
- 32x32 map (20x15 visible)
- 8x8 4bpp tiles
- overlapping maps
- shared tile set

Example Program

Tile Maps