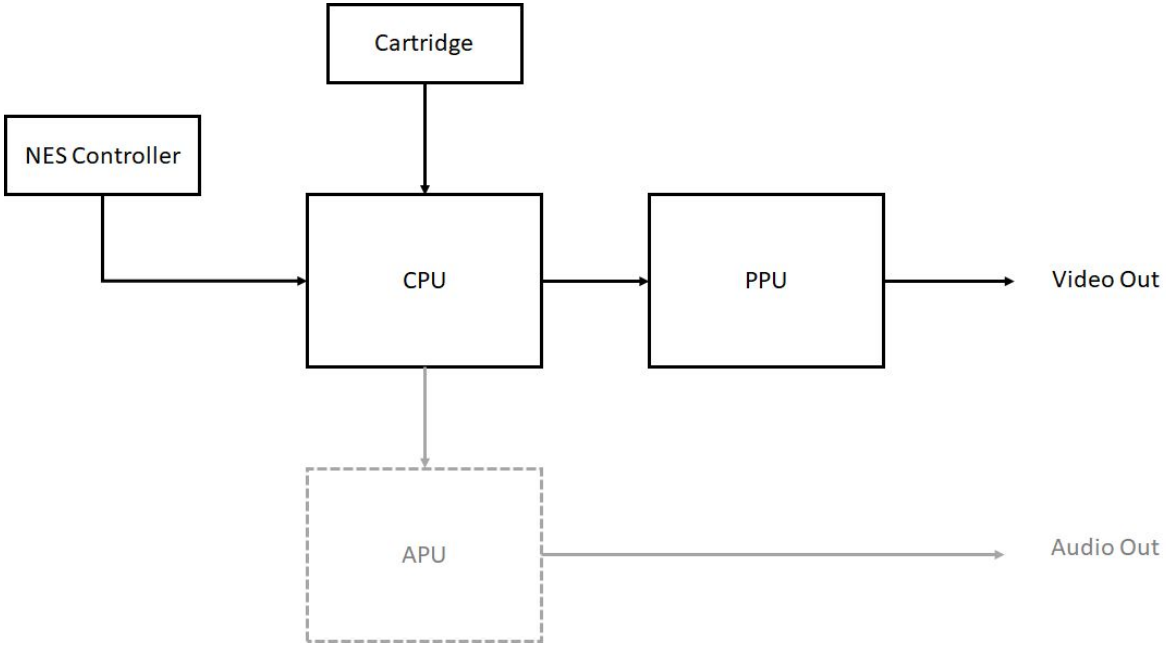
NES Hardware Emulation

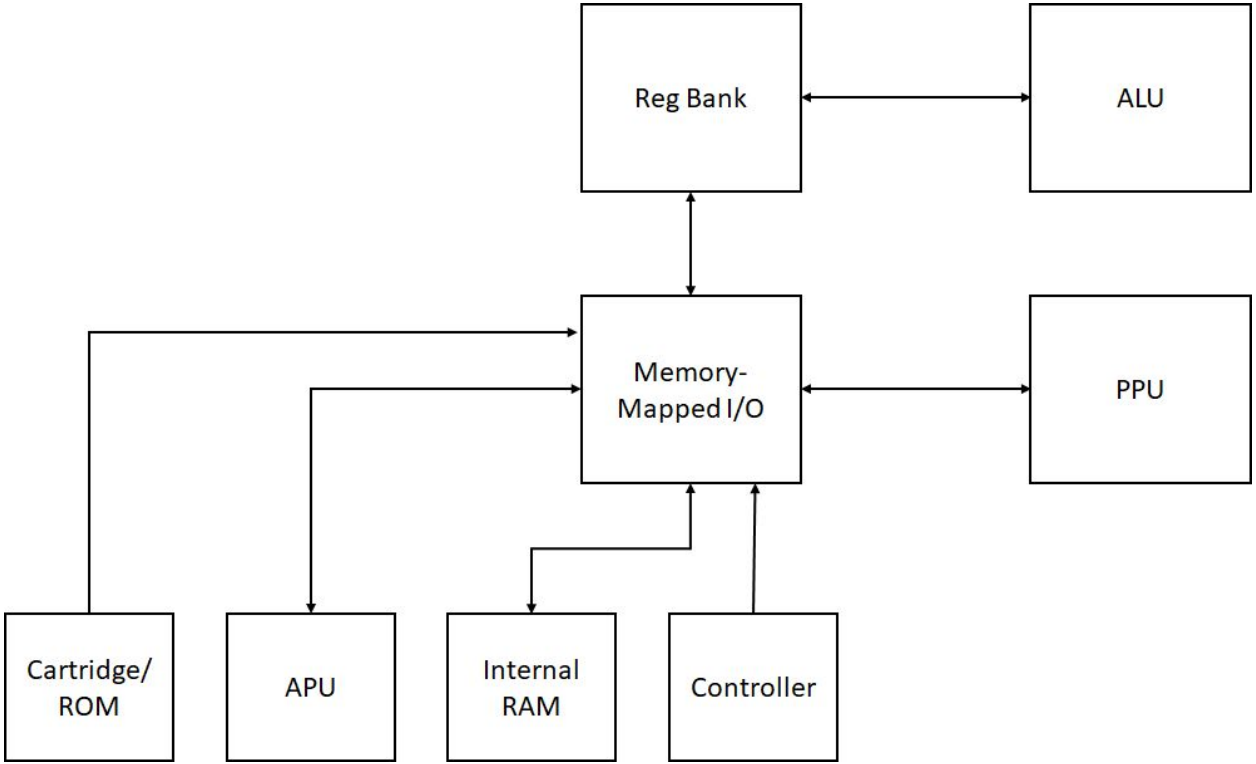
Sidne Gregory, Daniel Klahn, Israel Bonilla


Nintendo
ENTERTAINMENT SYSTEM
POWER
RESET
Nintendo
SELECT START

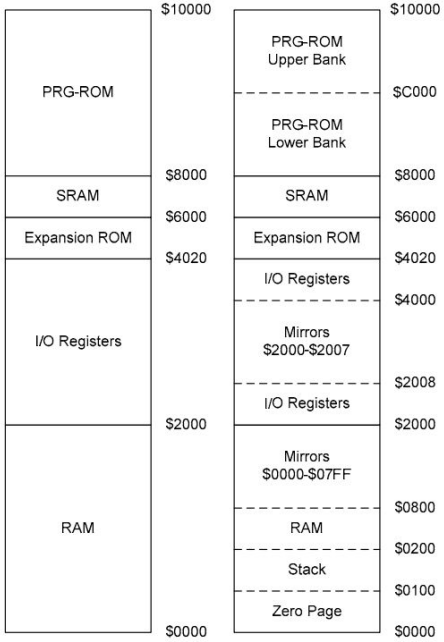
Overall System Architecture

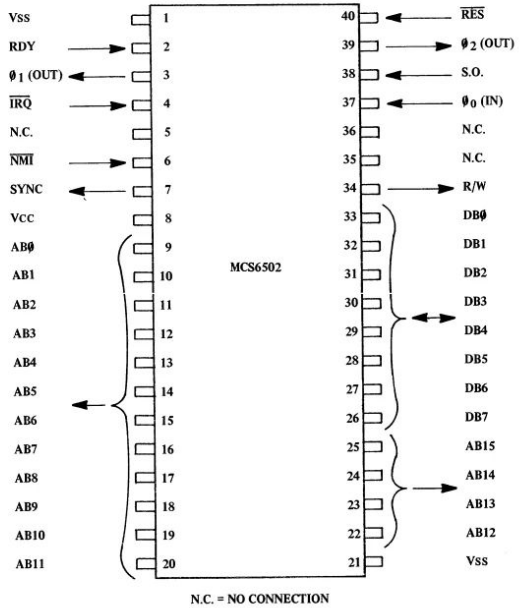


6502 Block Diagram



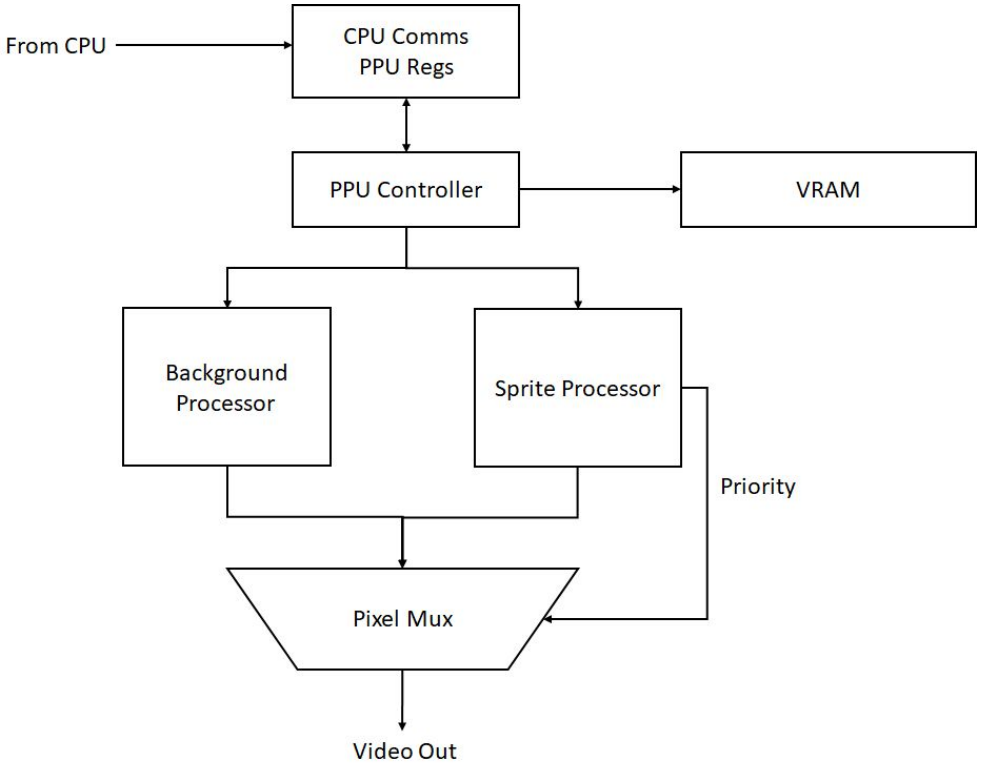
The CPU: MOS 6502 (Ricoh 2A03)





MCS6502 Pinout Designation
FIGURE 1.15

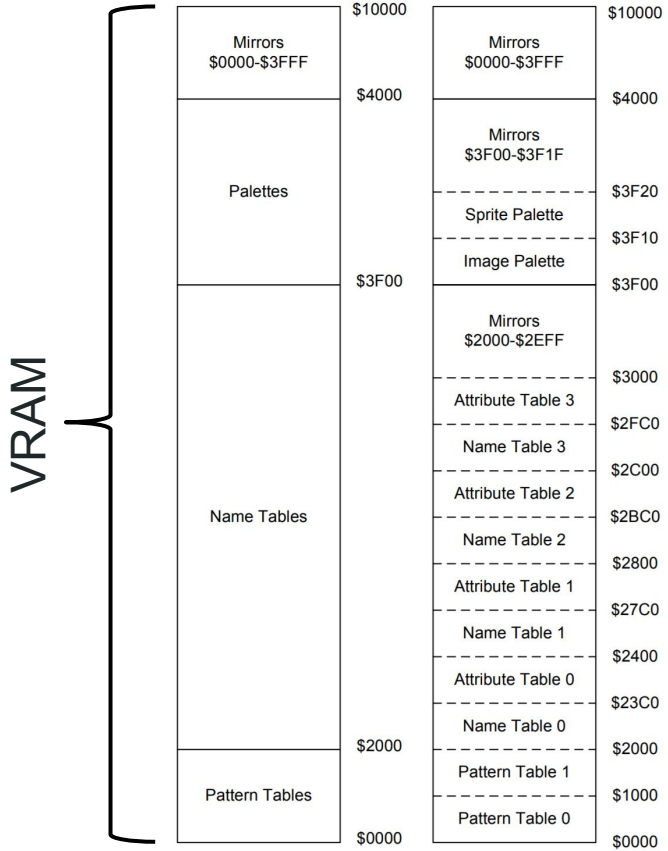
PPU Block Diagram



PPU Memory Model

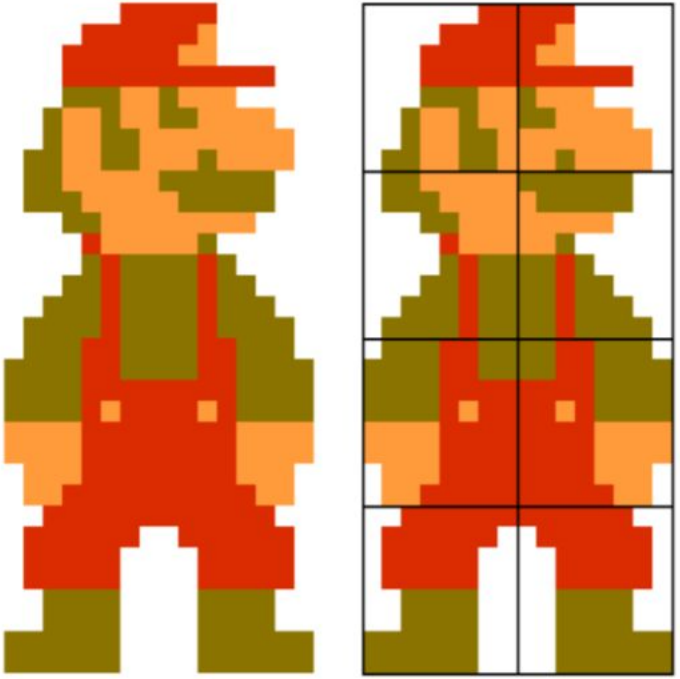
- Pattern tables: store the tiles
- Name tables: store the background
- Attribute tables: stores upper half of color IDs
- Color Palettes
- Sprite RAM: stores information about the sprites (internal to PPU)
    - Position
    - Layering
    - Orientation

- Memory mirroring



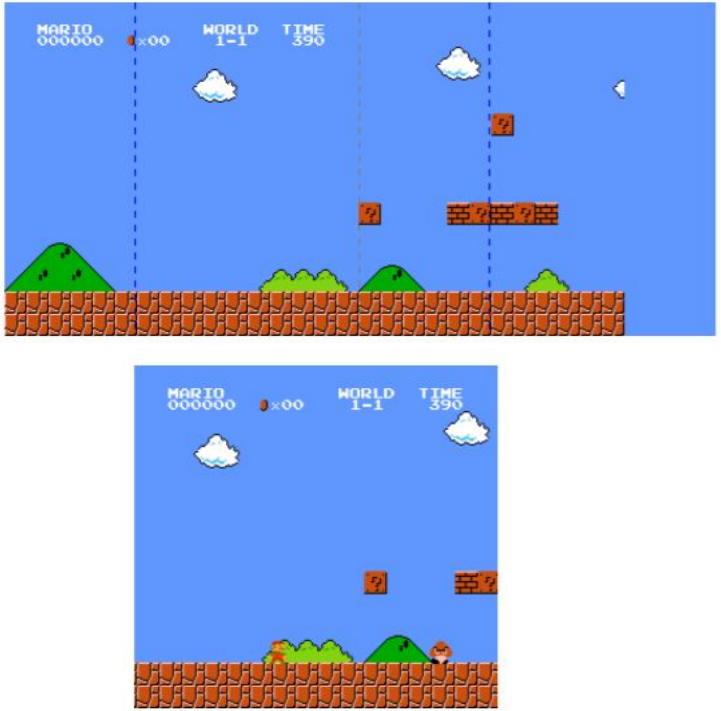
Sprite Processor

- Sprites are game objects that can move
- Sprite attributes include a tile ID, position
- The PPU can store a maximum of 64 sprites in sprite RAM (aka Object Attribute Memory or OAM)
- The sprite layering is determined by their order in OAM



Background Processor

- Background is drawn by mapping to 8x8 tiles stored in VRAM (Video RAM)
- Name tables store which tile is used where
- Attribute tables hold part of the color information for the specific tile
- Enough storage for 2 name tables to allow for scrolling



Goals

Minimum Viable Product: Show CPU reading BRAM and executing a subset of its instruction set. Show PPU displaying sprites and background in testbench

Base: Integrated CPU and PPU that can run select NES game ROMs from BRAM and allow play with controller

Stretch: Include APU for sound with gameplay, Interfacing with actual NES cartridges

Project Timeline