Team Nintendo®

6.111 Final Project

Jonathan Downey
Lauri Kauppila
Brian Myhre

Project Motivation

- Distinct Sub-systems
- Complex Behavior and Architecture
- FPGA Required
  - Not Possible with Microcontroller
  - Large Amount of Logic
- Entertainment Value
- Appreciated by Non-Technical People

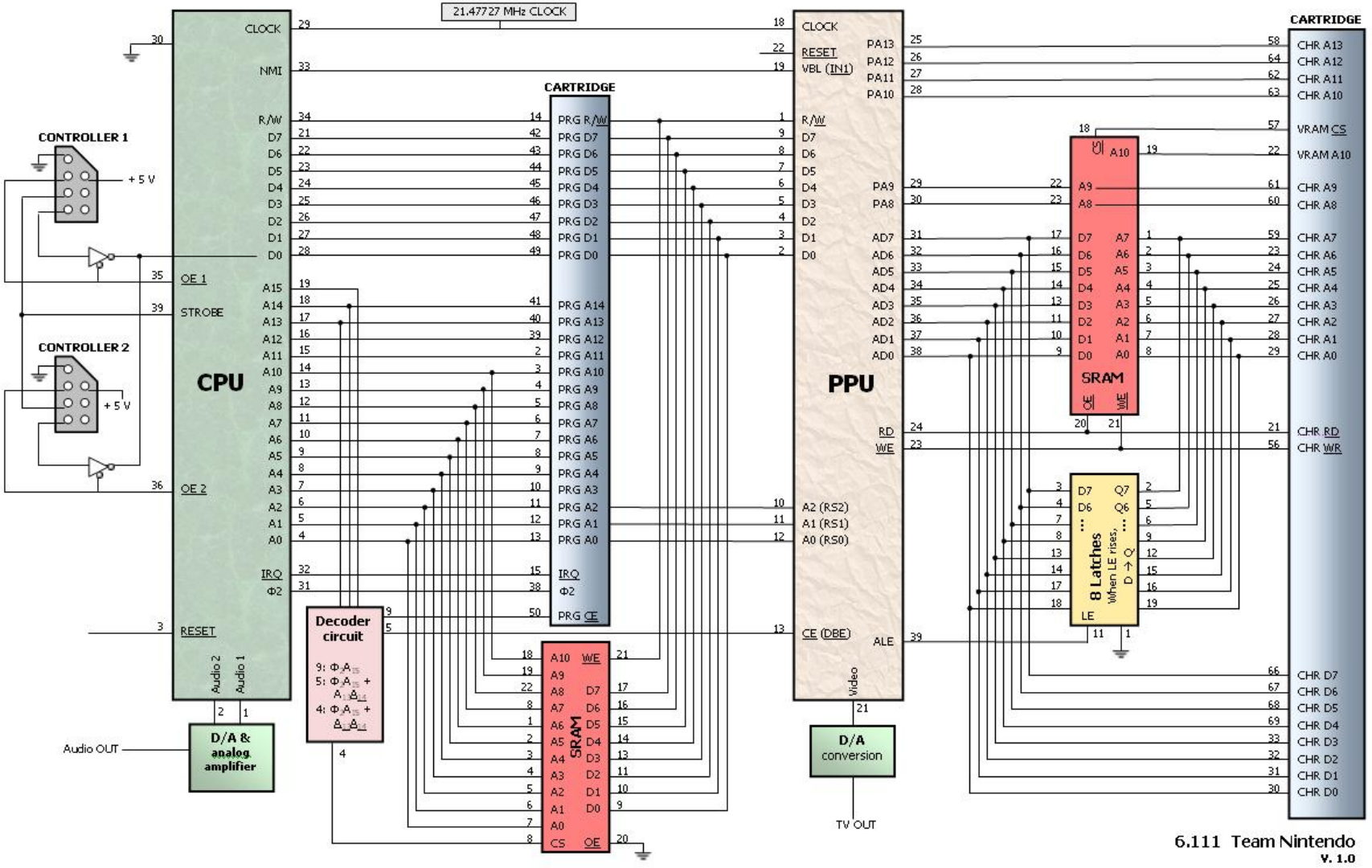
NES Background

- 1983, NES Introduced in Japan
- 1985, NES released in US
- 1995, Discontinued Production of NES
- Nintendo sold over 62 million NES systems and 500 million games
- Currently the most widely emulated system with over thirty different emulators.

NES System Overview

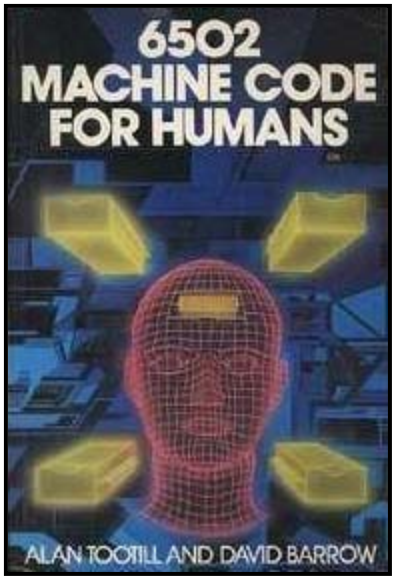
- CPU: Custom 8-bit 2A03
  - Based on the 6502 core running at 1.79 MHz
  - Onboard Audio: 2 Square Waves, 1 Triangle, 1 Digital, 1 Noise
  - Input from Control Pads
- Main Ram: 2 KB
- PPU: Picture Processing Unit
  - Sprite and background generation
  - 2 KB Video Ram
  - 64 Colors
  - 256 by 240 pixel resolution
- Removable Game Cartridges
  - Program Data ROM: 32KB to 1MB
  - Graphical Data ROM: 8KB to 1MB

Schematic


21.47727 MHz CLOCK
CPU
PPU
CARTRIDGE
CONTROLLER 1
CONTROLLER 2
+5 V
CLOCK
NMI
R/W
D7
D6
D5
D4
D3
D2
D1
D0
OE 1
OE 2
STROBE
RESET
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
IRQ
Φ2
Audio 2
Audio 1
Audio OUT
D/A & analog amplifier
Decoder circuit
SRAM
PRG R/W
PRG D7
PRG D6
PRG D5
PRG D4
PRG D3
PRG D2
PRG D1
PRG D0
PRG A14
PRG A13
PRG A12
PRG A11
PRG A10
PRG A9
PRG A8
PRG A7
PRG A6
PRG A5
PRG A4
PRG A3
PRG A2
PRG A1
PRG A0
PRG CE
RESET
VBL (IN1)
PA13
PA12
PA11
PA10
PA9
PA8
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
RD
WE
A2 (RS2)
A1 (RS1)
A0 (RS0)
CE (DBE)
ALE
Video
D/A conversion
TV OUT
8 Latches
When LE rises, D → Q
LE
CHR A13
CHR A12
CHR A11
CHR A10
VRAM CS
VRAM A10
CHR A9
CHR A8
CHR A7
CHR A6
CHR A5
CHR A4
CHR A3
CHR A2
CHR A1
CHR A0
CHR RD
CHR WR
CHR D7
CHR D6
CHR D5
CHR D4
CHR D3
CHR D2
CHR D1
CHR D0
6.111 Team Nintendo
v. 1.0

CPU: The Historic 6502

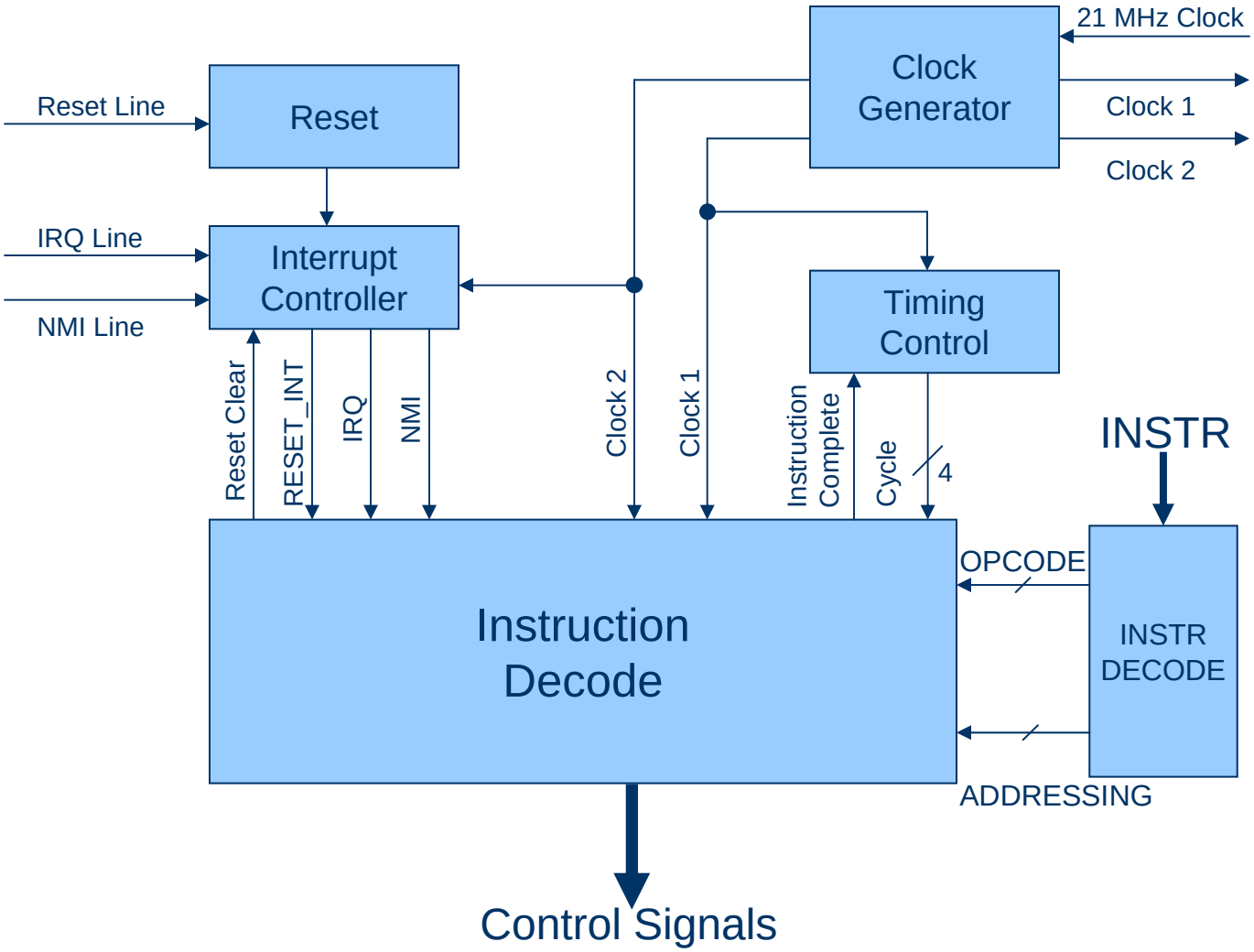
- Designed by MOS Technologies, a division of Commodore
- Released in 1975 for $25
- In the first PCs such as the Commodore 64, Apple I and Apple II.
- Landmark in Microprocessor design.
  - New “pipelining” led to no wasted memory cycles and amazing speed.
  - Fast Access to RAM, less registers
- Efficient design of the 6502 is said to have inspired the development of the ARM processors



6502 Specifics

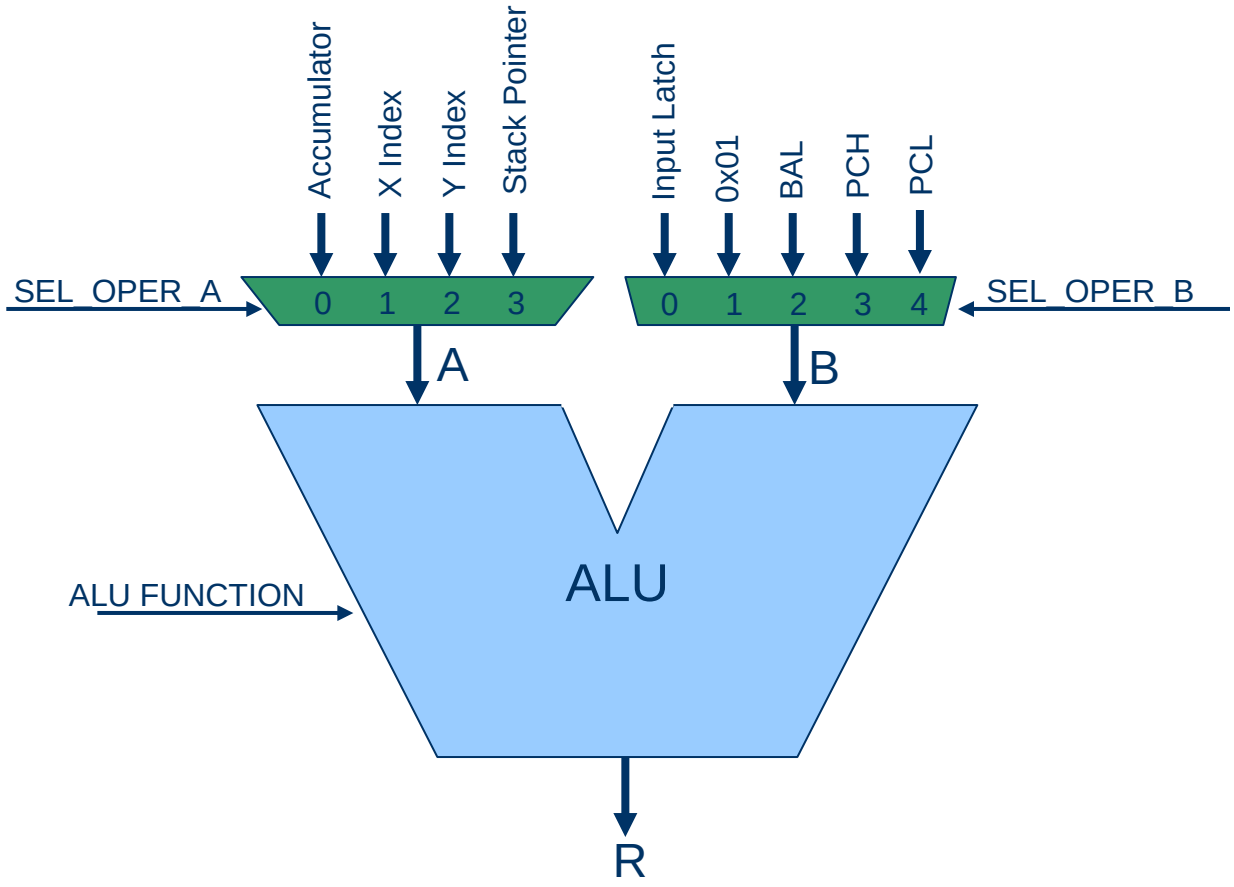
- 8-bit Bi-directional Data Bus
- 16-bit Address Bus (64 KB)
- Complex Instruction Set with over 50 Instructions
- Eleven Addressing Modes
- Operation up to 2 MHz

CPU Control

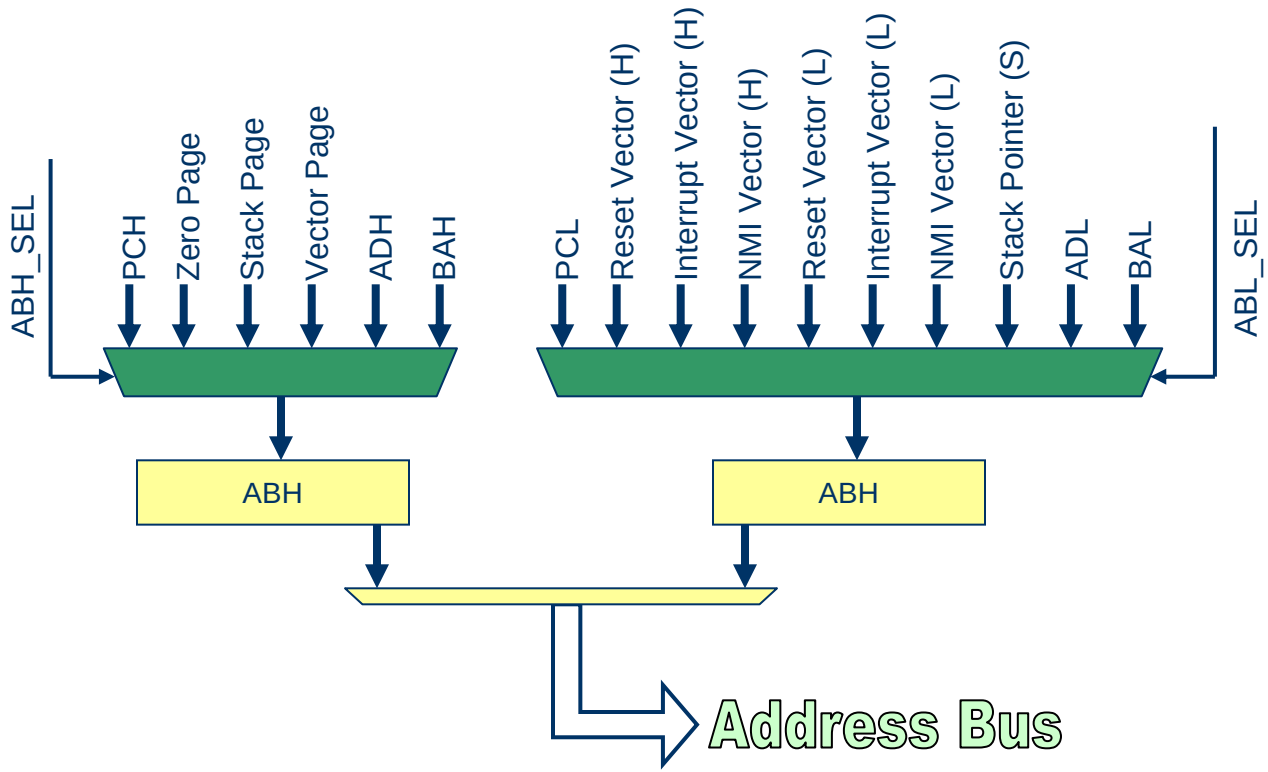

21 MHz Clock
Clock Generator
Clock 1
Clock 2
Reset Line
Reset
IRQ Line
NMI Line
Interrupt Controller
Timing Control
Reset Clear
RESET_INT
IRQ
NMI
Clock 2
Clock 1
Instruction Complete
Cycle
4
INSTR
OPCODE
INSTR DECODE
Instruction Decode
ADDRESSING
Control Signals

Arithmetic Logic Unit

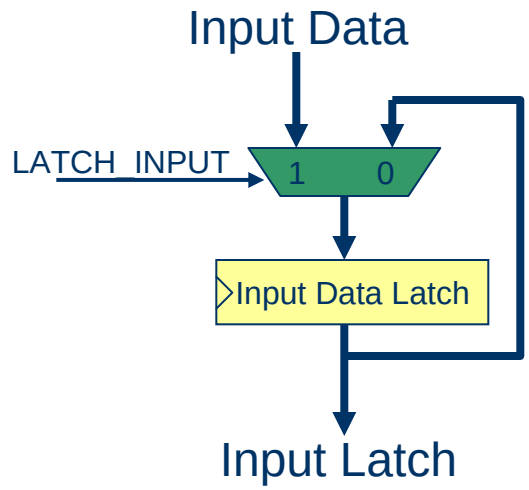
ALU Functions:
0: A
1: B
2: Add
3: Subtract
4: AND
5: OR
6: XOR
7: Shift A Left
8: Shift A Right
9: Rotate A Left
10: Rotate A Right
11: Shift B Left
12: Shift B Right
13: Rotate B Left
14: Rotate B Right

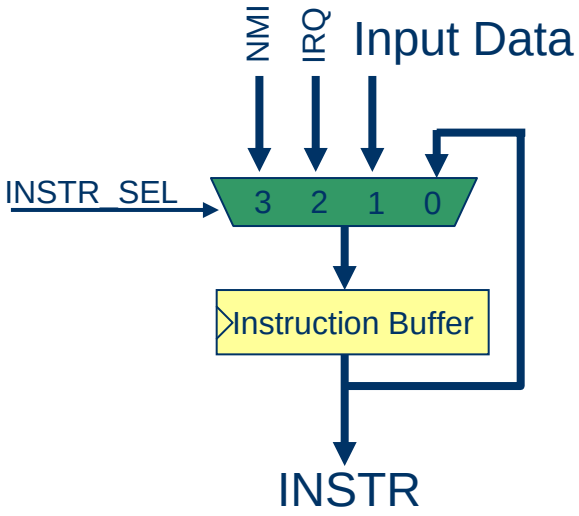


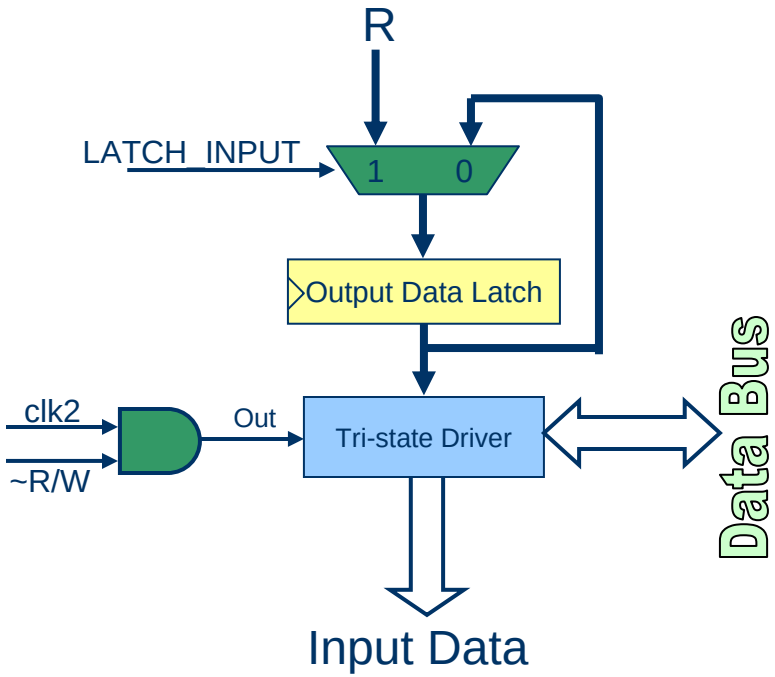
Address Bus


ABH_SEL
PCH
Zero Page
Stack Page
Vector Page
ADH
BAH
PCL
Reset Vector (H)
Interrupt Vector (H)
NMI Vector (H)
Reset Vector (L)
Interrupt Vector (L)
NMI Vector (L)
Stack Pointer (S)
ADL
BAL
ABL_SEL
ABH
ABH
Address Bus

Data Bus


Input Data
LATCH_INPUT
1
0
Input Data Latch
Input Latch


NMI
IRQ
Input Data
INSTR_SEL
3
2
1
0
Instruction Buffer
INSTR


R
LATCH_INPUT
1
0
Output Data Latch
clk2
~R/W
Out
Tri-state Driver
Data Bus
Input Data

Picture Processing Unit

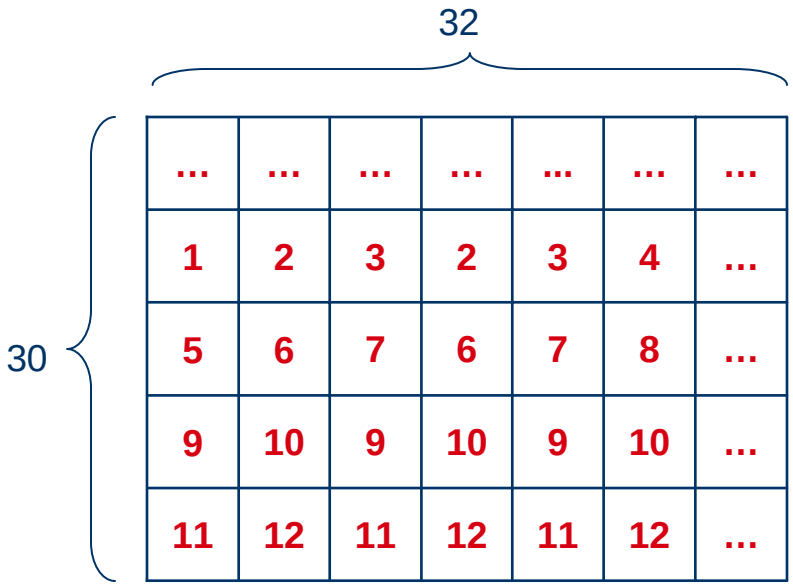
- Works 3 times faster than CPU, a whopping 5.37 MHz
- 256 x 240 resolution
- 64 colors
- Contains:
  - ~300 bytes of internal RAM
  - 2kb of external RAM (inside NES)
  - 8kb or more of external ROM (inside cartridge)

Background Rendering

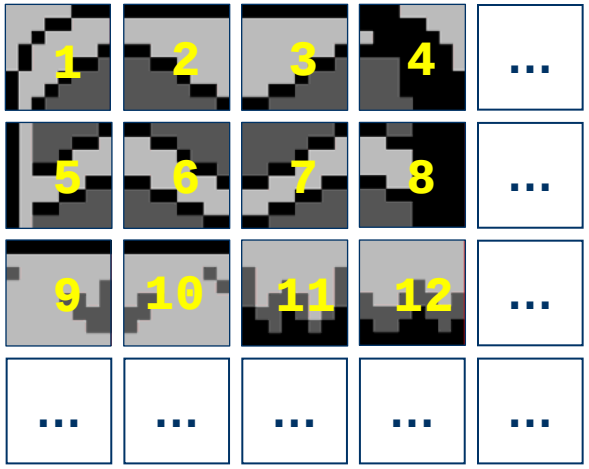
- 32 x 30 “tiles” in one screen
- Each tile contains 8x8 pixels
- Background rendered ~ in real time
- Four steps to picture creation:

1. Fetch a pattern for the tile (no color information yet)
2. Fetch an “attribute” for the tile (tells what range of colors exist)
3. Fetch colors from a color table
4. Output pixel

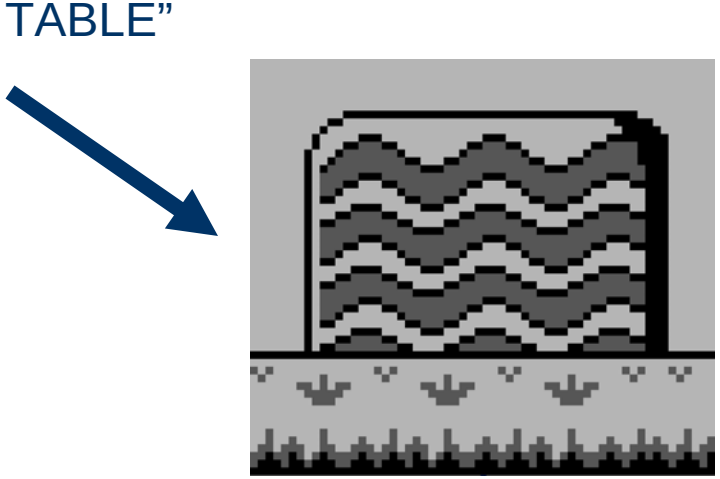
Example



“NAME TABLE”

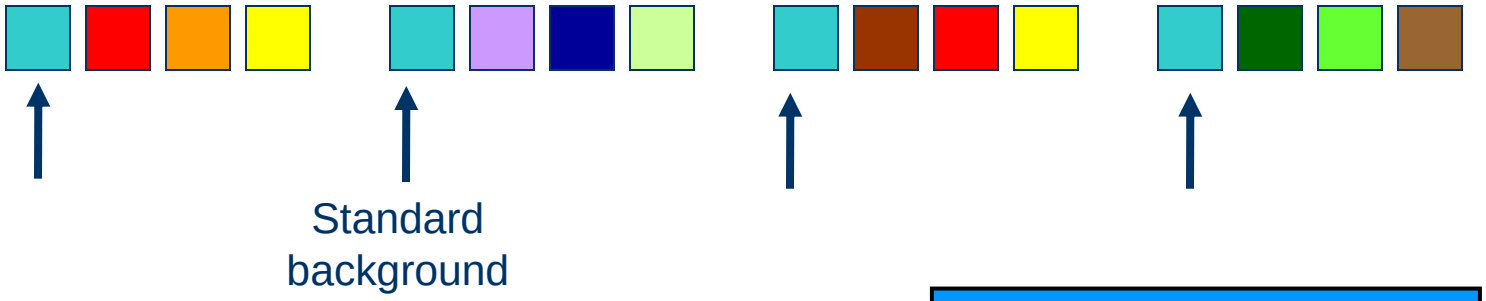


“PATTERN TABLE”

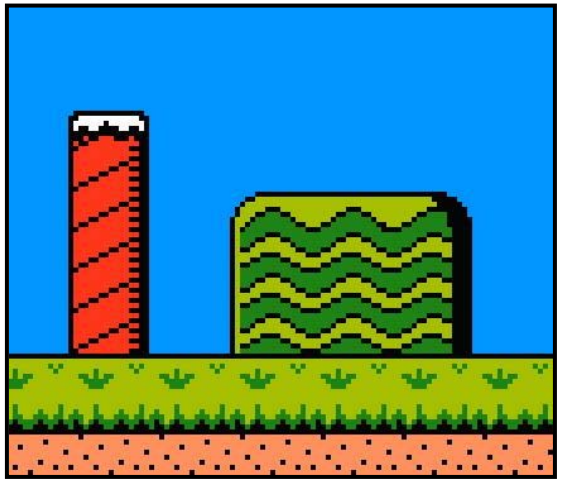


Coloring

- 4 palettes for background. Each palette has 3 unique colors and a standard background color



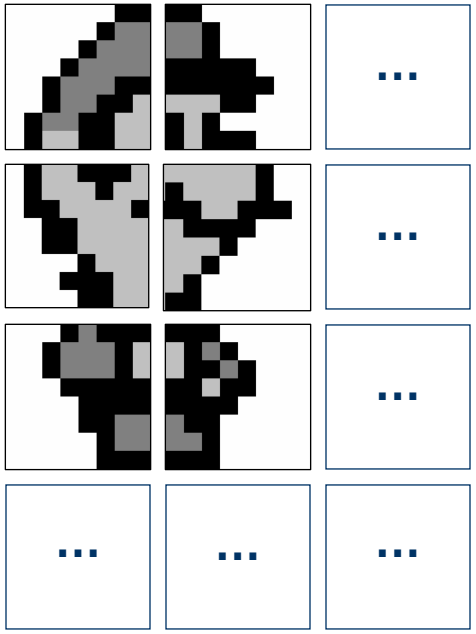
- Each 2 x 2 tile area uses the same palette. This “palette select” data is stored in an “attribute table”.
- 4-bit pattern data determines what color to choose inside a palette



Sprite Rendering

- 64 sprites (8x8 pixel tiles) can be shown per frame. 8 sprites per line (due to time constraints)
- Sprite RAM (inside PPU) contains:
  - 8-bit x-coordinate
  - 8-bit y-coordinate
  - Vertical/horizontal flip
  - Pattern table address
  - 2-bit palette selection

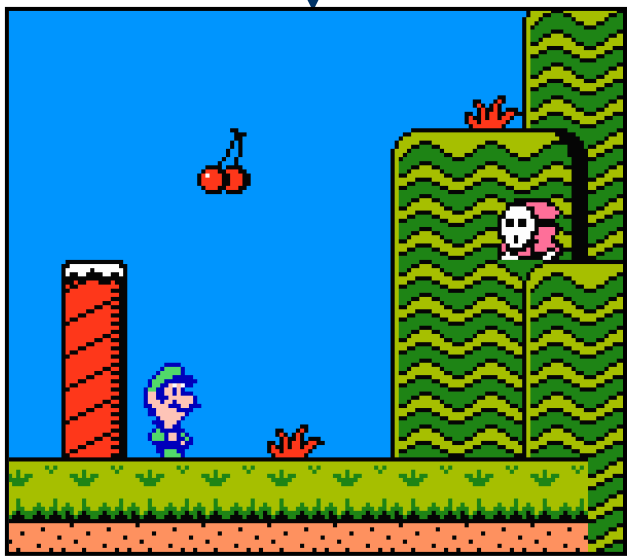
Example



“PATTERN TABLE”

Sprite memory

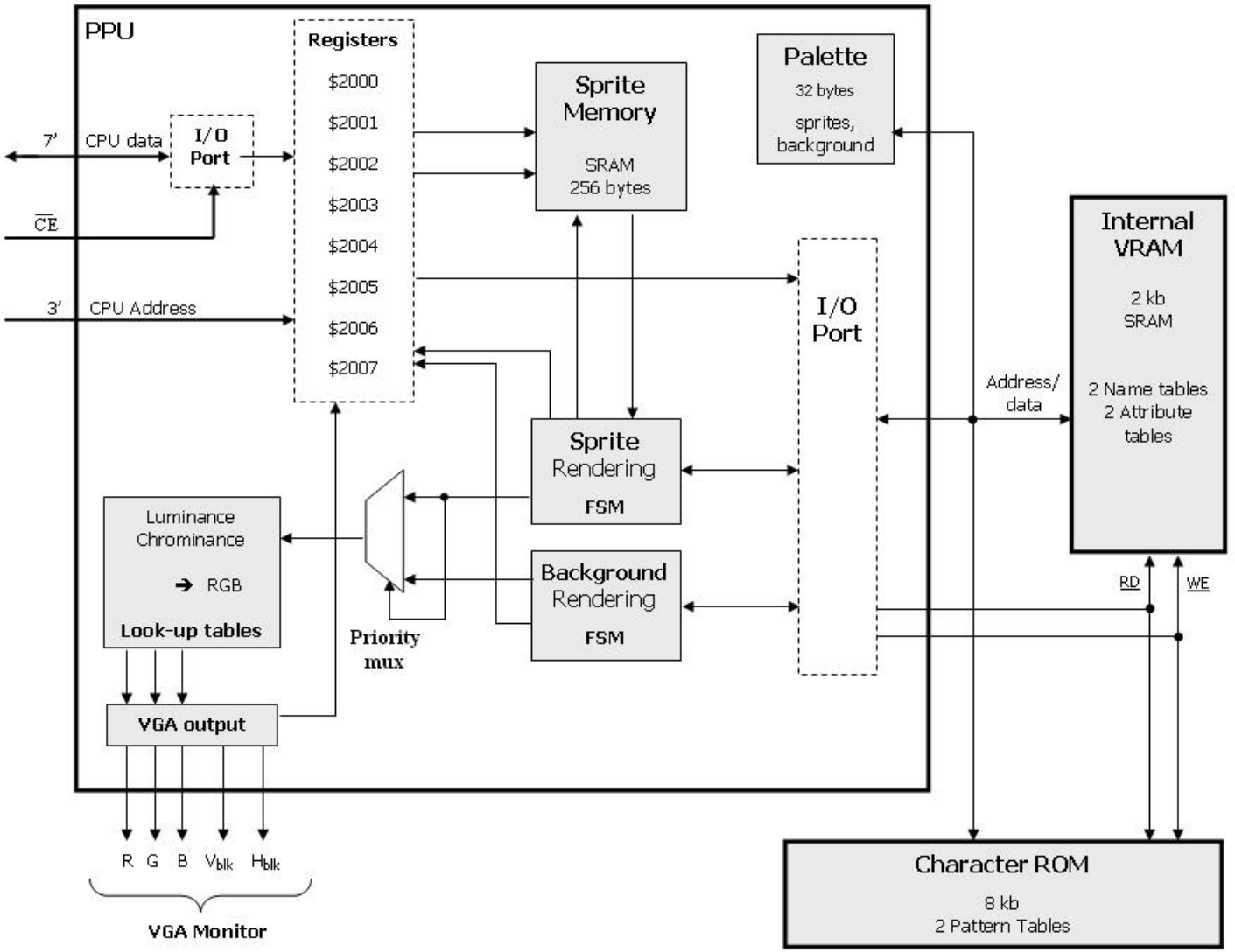
- coordinates
- pattern
- attributes (flip, color)



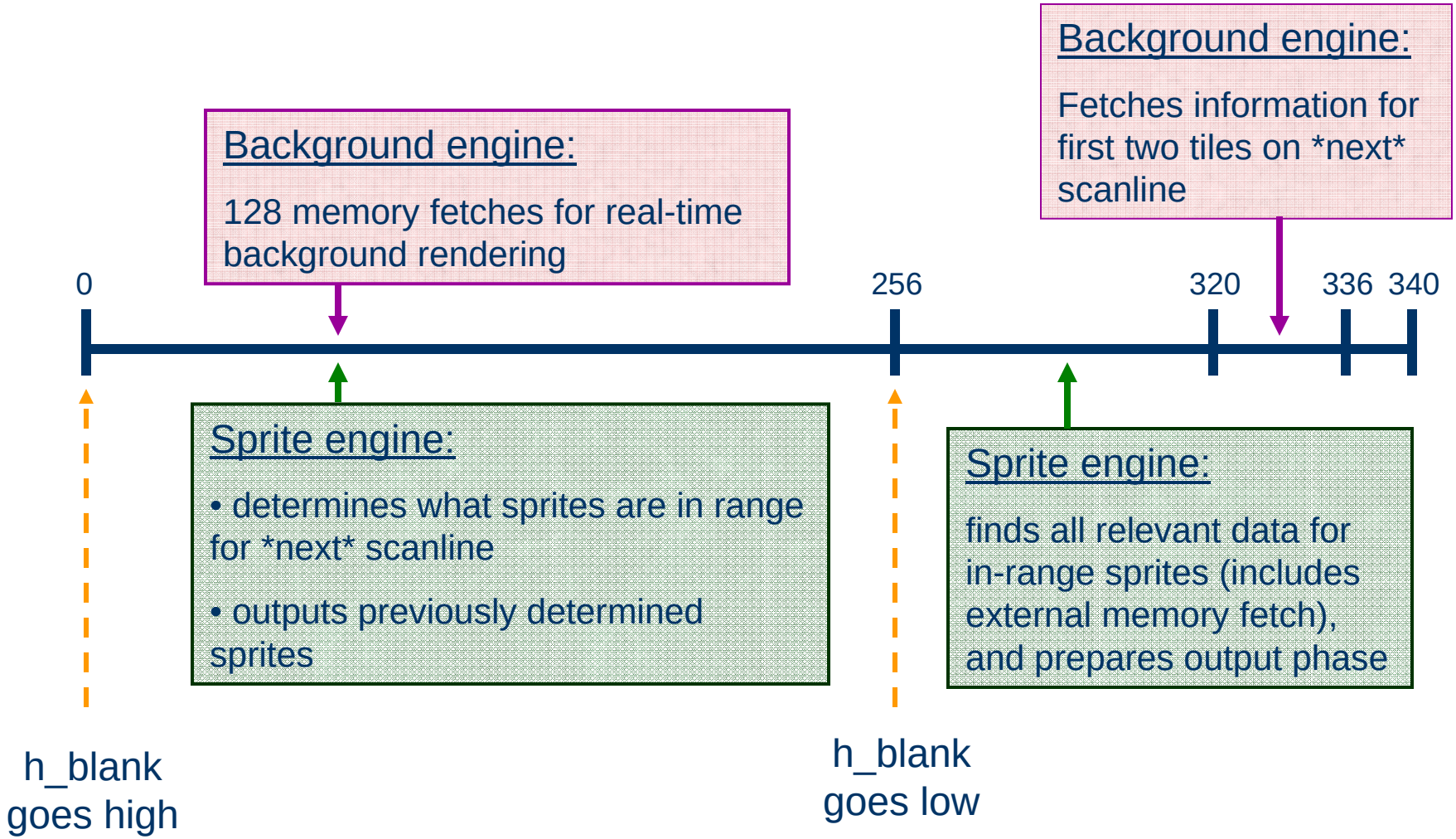
Final Output

- Background and sprite pixels are overlaid, with specific priority instructions
- Send CPU a “hit flag” for object collision, “more-than-8-sprites-on-one-line flag”

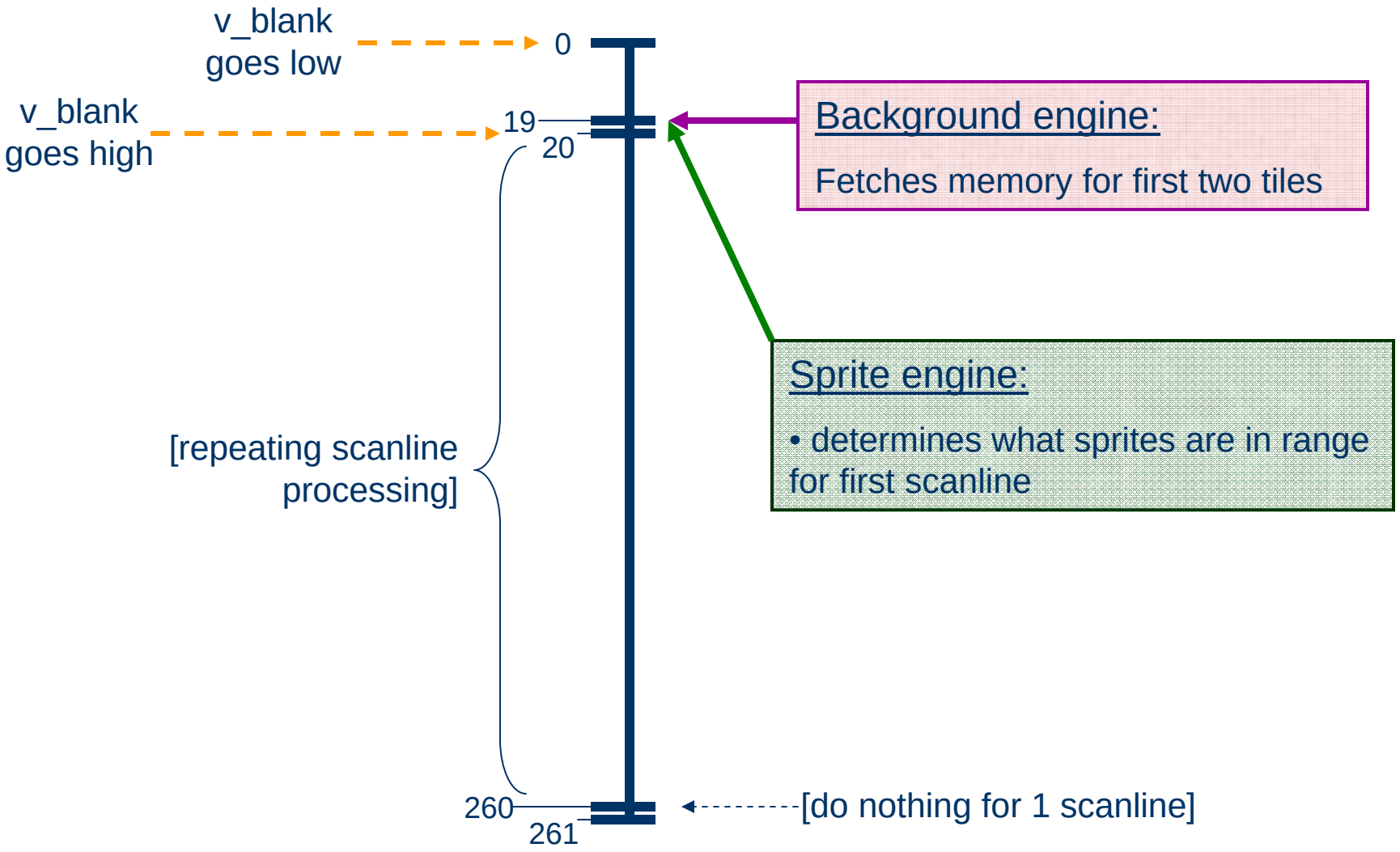
PPU Architecture



PPU Timing / Memory Access



PPU Timing / Memory Access



Possible Extensions

- Audio Processing Unit
- Ability to Load Multiple Games

Team Nintendo®

6.111 Final Project

Jonathan Downey
Lauri Kauppila
Brian Myhre