Space Invaders on FPGA with WII-remote

Soclab Project

Introduction

- Space Invaders
- Wii-remote

Memories

- ROM
  - Sprites
  - Color palette
- RAM
  - Enemy locations
  - Bullet locations

ROM

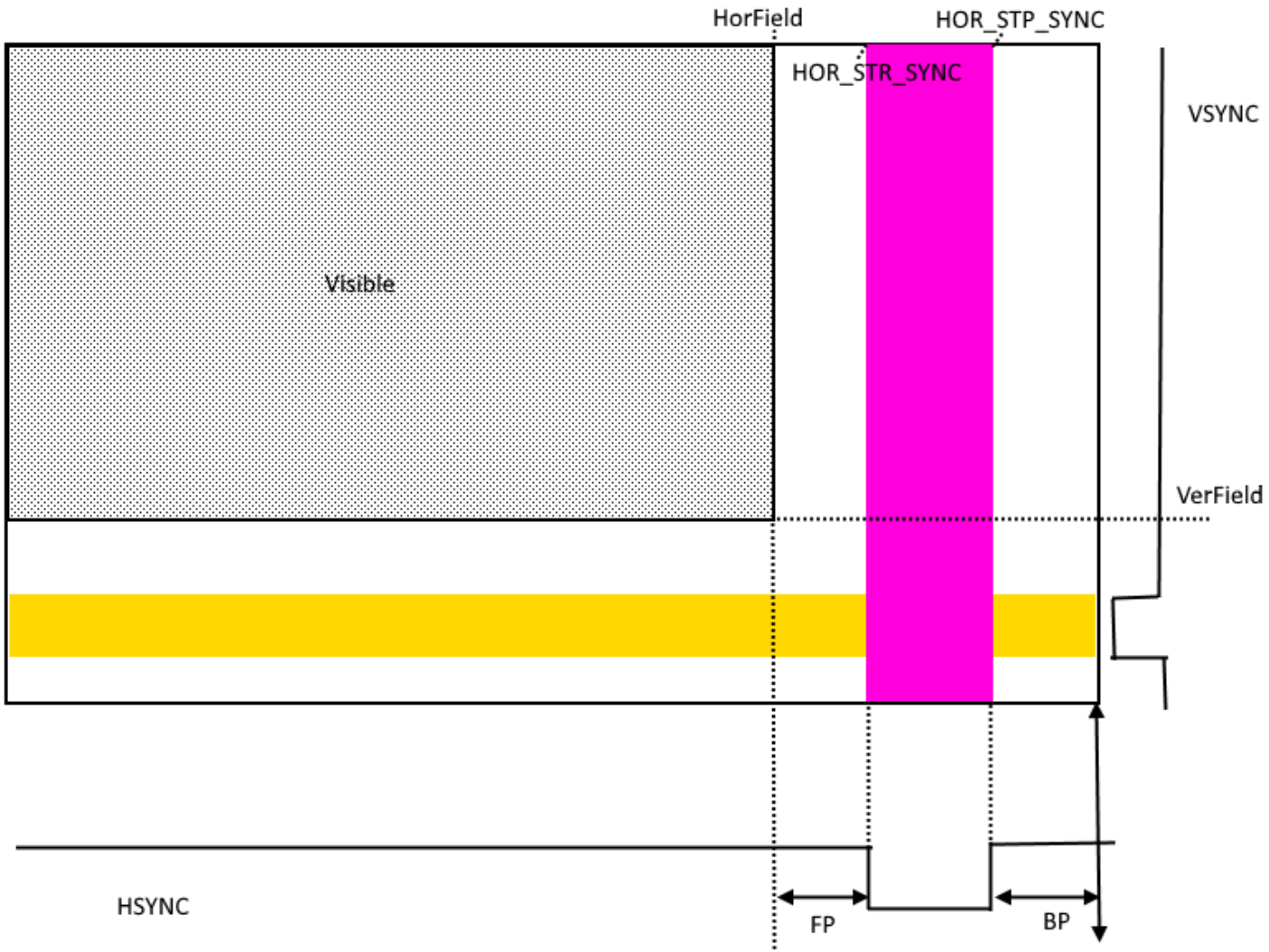
- 1-Port ROM (color palette)
  - 4-bit addresses
  - 24-bit words (8bits for R, G & B)
  - Keeps the sprite memories small
- 1-Port ROM 3x (sprite ship, bullet, enemy)
  - 8-bit addresses (16x16px)
  - 4-bit words
    - will be resolved to 24-bit color in color palette

```
ship_image image(.address(address), .clock(clock), .q(color_address));
colorpalette cp(.address(color_address), .clock(clock), .q(color));
```

RAM

- 2-Port RAM (bullets)
  - 6-bit addresses
  - 24-bit words ({y, x, active})
    - Active-bit is for control
- 2-port RAM (enemies)
  - Idem bullets

Logic


HorField
HOR_STP_SYNC
HOR_STR_SYNC
VSYNC
Visible
VerField
HSYNC
FP
BP

Logic

- !calc
  - All pixels are calculated one by one
  - Compare for collision
    - If collision write 0 to address
      - This is why 2-port rams are handy
  - Draw if active-bit is 1
  - If something from memory is drawn
    - get next from memory
      - Sorted memory from left to right and top to bottom
      - {y, x}
  - Else background
    - LFSR red & white for moving effect

Logic

- calc - statemachines
  - Remove empty memory slots
    - 1,2,0,4,5 -> 1,2,4,5,0
  - Insert/create new memory entries
    - Insert in order
    - 3 -> 1,2,4,5,0
    - 1,2,3,4,5
  - Move memory entries
  - Idle
    - Nothing left to be done wait for next calc

Demonstration

Demonstration