

# Ultrasonic Fruit Ninja on DE1-SoC

Team 3 (Peiheng Li, Chengrui Li, Yitong Bai) | CSEE 4840 Embedded Systems Design

Platform: Cyclone V SE 5CSEMA5F31C6

# Presentation Outline

1

## Project Overview

High-level gameplay and system objectives.

2

## System Architecture

The complete hardware/software map.

3

## Hardware Fabric

Sensor driver and zero-frame-buffer VGA.

4

## MMIO Interface

Avalon-MM registers and bridging.

5

## Software Layer

Game logic and Linux kernel abstractions.

6

## Results & Demo

Resource utilization and live gameplay.

# Project Overview & Gameplay

## Core Mechanic

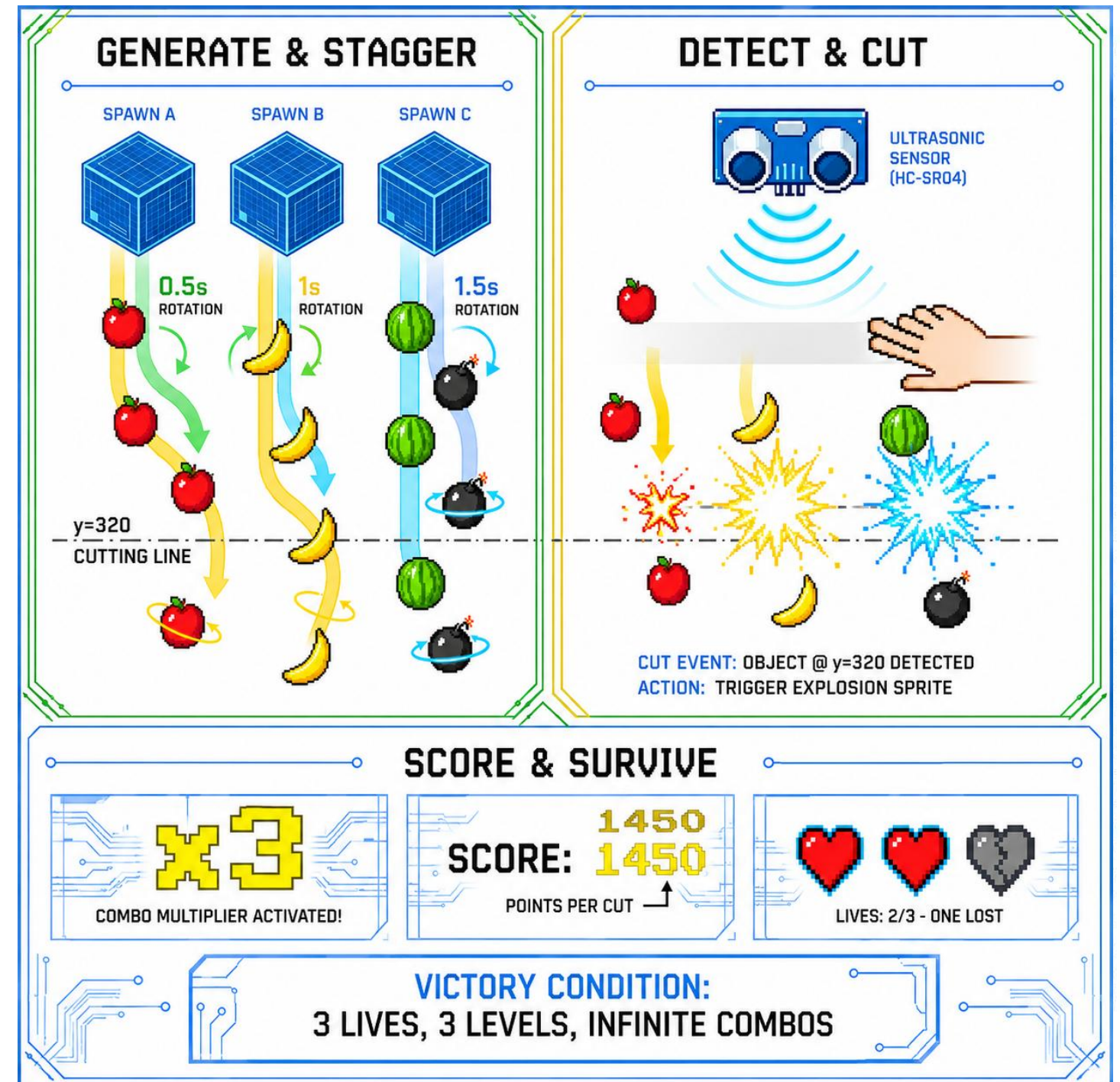
- Wave hand over the sensor to trigger a cut along the dashed horizontal screen line.

## Rules & Scoring

- +1 Point for Cut Fruit
- -1 Life for Cut Bomb
- -1 Life for Missed Fruit

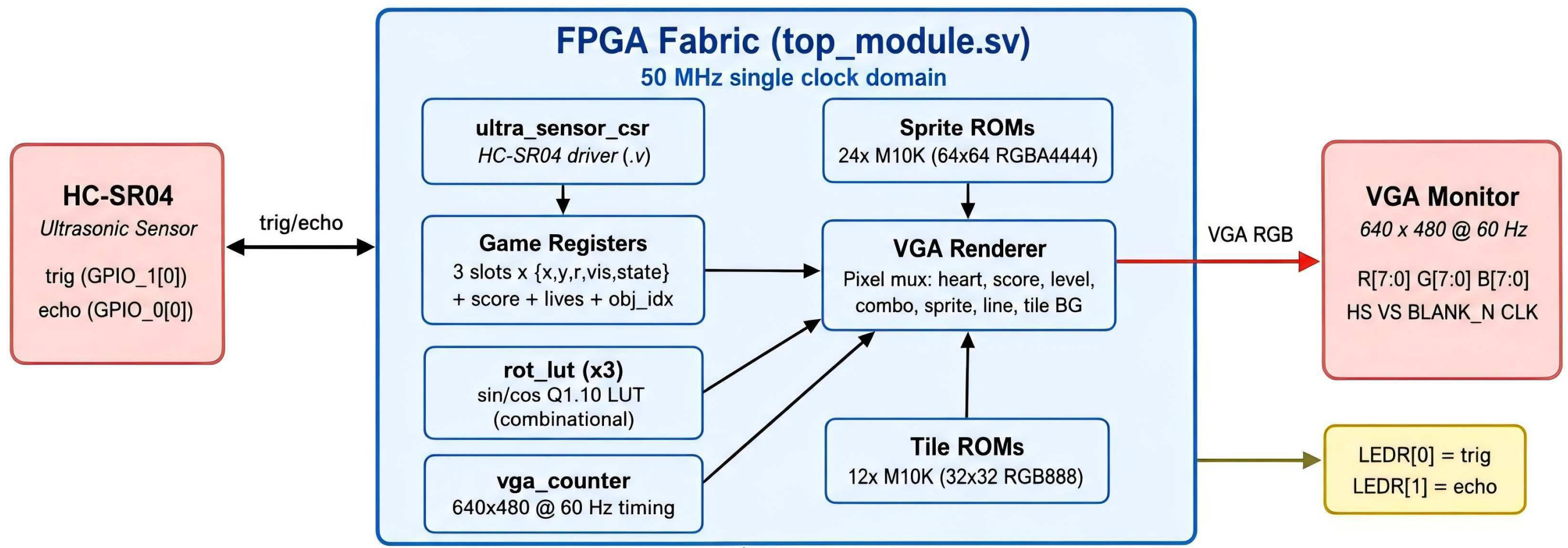
## Progression

- Multi-slot scaling (Level 1 to 3 increases concurrent objects).
- Consecutive cuts build **Combo multipliers** to heal lost lives.

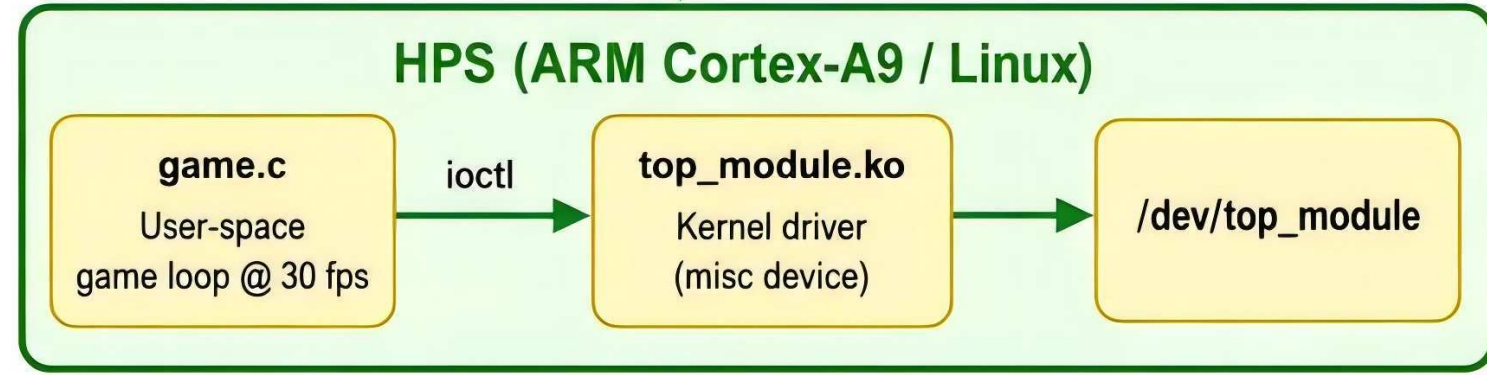


# The Full-Stack Architecture

**Hardware Fabric**  
Sensor driver, ROMs, and VGA renderer running synchronously at 50 MHz.



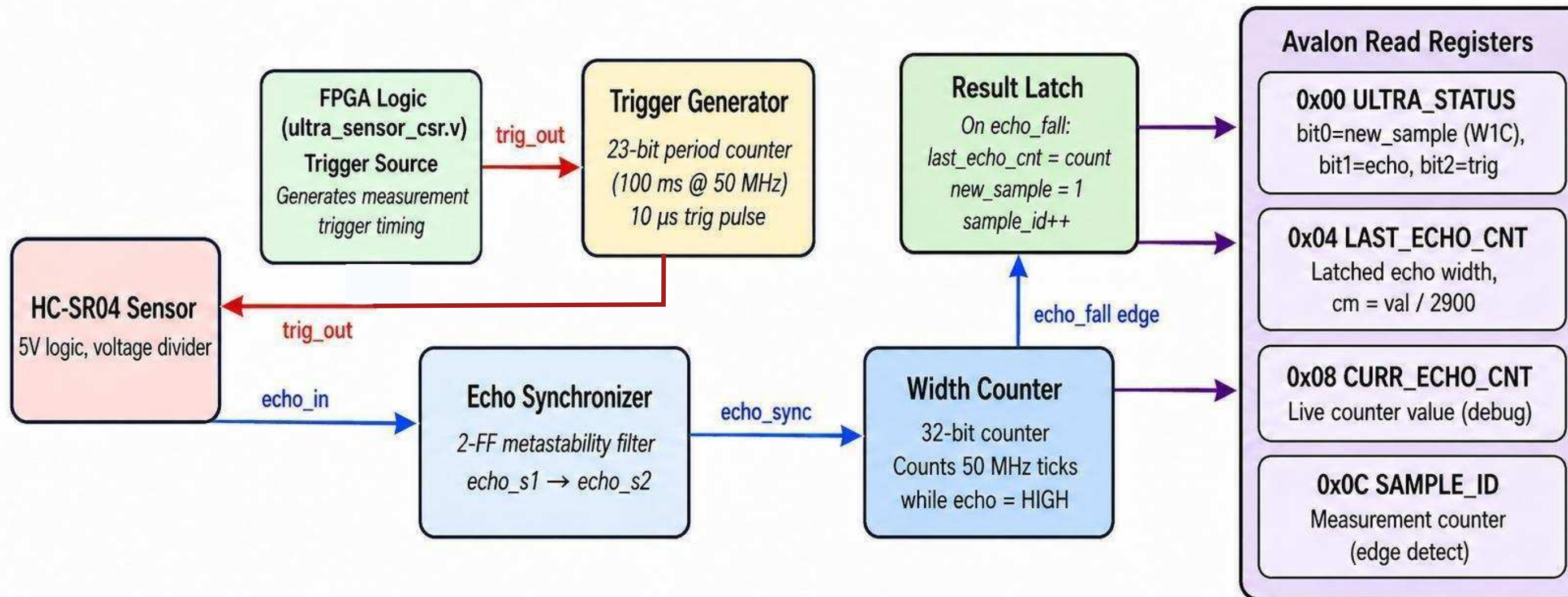
Avalon-MM  
(h2f\_lw\_axi\_master @ 0xFF200000)



**Software Space**  
ARM Cortex-A9 running the 30 FPS physics loop via ioctl commands.

# Hardware Layer: Ultrasonic Sensor Driver

## (C) HC-SR04 Ultrasonic Sensor Driver (ultra\_sensor\_csr.v)



### Measurement Timing

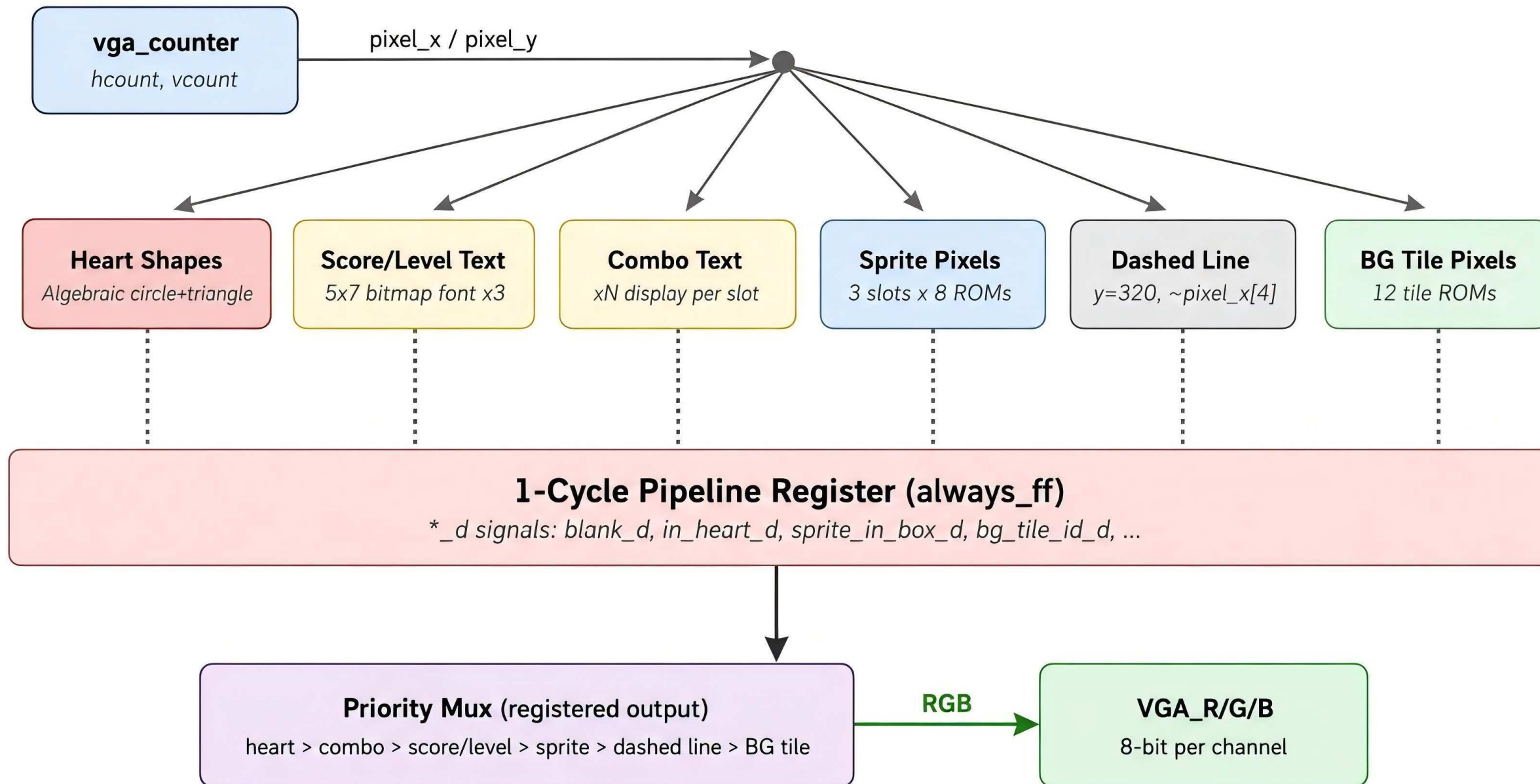
Period  $\approx$  100 ms (10 Hz) | Echo pulse: 150 μs - 25 ms | Distance:  $cm \approx ticks / 2900$

### Software Consumption (game.c)

1. MOTION\_READ ioctl reads all 4 regs
2.  $cm = changed\_pixel\_count / ULTRA\_TICKS\_PER\_CM$
3. Edge detect: `!prev_near && (cm < 20) → swipe`

# Hardware Layer: Zero-Framebuffer VGA Pipeline

(A) VGA Pixel Rendering Pipeline — per-pixel combinational + 1-cycle registered output



## 1-Cycle Pipeline

No frame buffer SDRAM. Every pixel is calculated on the fly based on current (**pixel\_x**, **pixel\_y**) coordinates.

## Priority Multiplexer

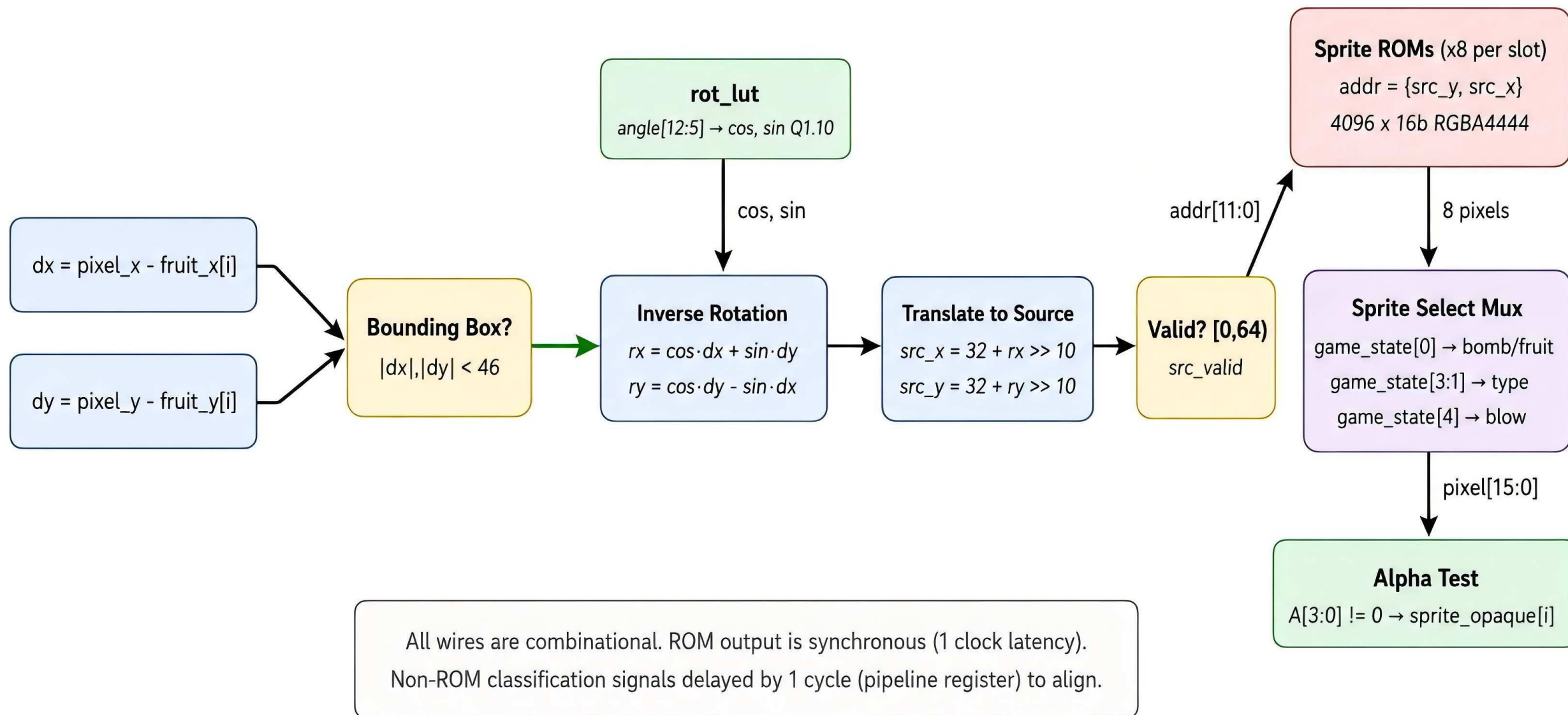
Strict drawing order enforced combinatorially: Hearts > Combo > Score > Sprite > Dashed Line > Background Tile.

## DAC Output

Registered 8-bit per channel RGB fed directly to the ADV7123 DAC at 60Hz.

# Hardware Layer: Sprite Rotation Engine

(B) Sprite Rotation Engine — inverse rotation sampling (x3 instances, combinational)



## Mathematical Operations

### Inverse Rotation Sampling

Maps screen pixels back to the source texture rather than pushing texture pixels to the screen to avoid holes.

### Fixed-Point Math

An 8-bit angle queries a Q1.10 signed sin/cos quarter-wave Lookup Table (65 entries), avoiding floating point.

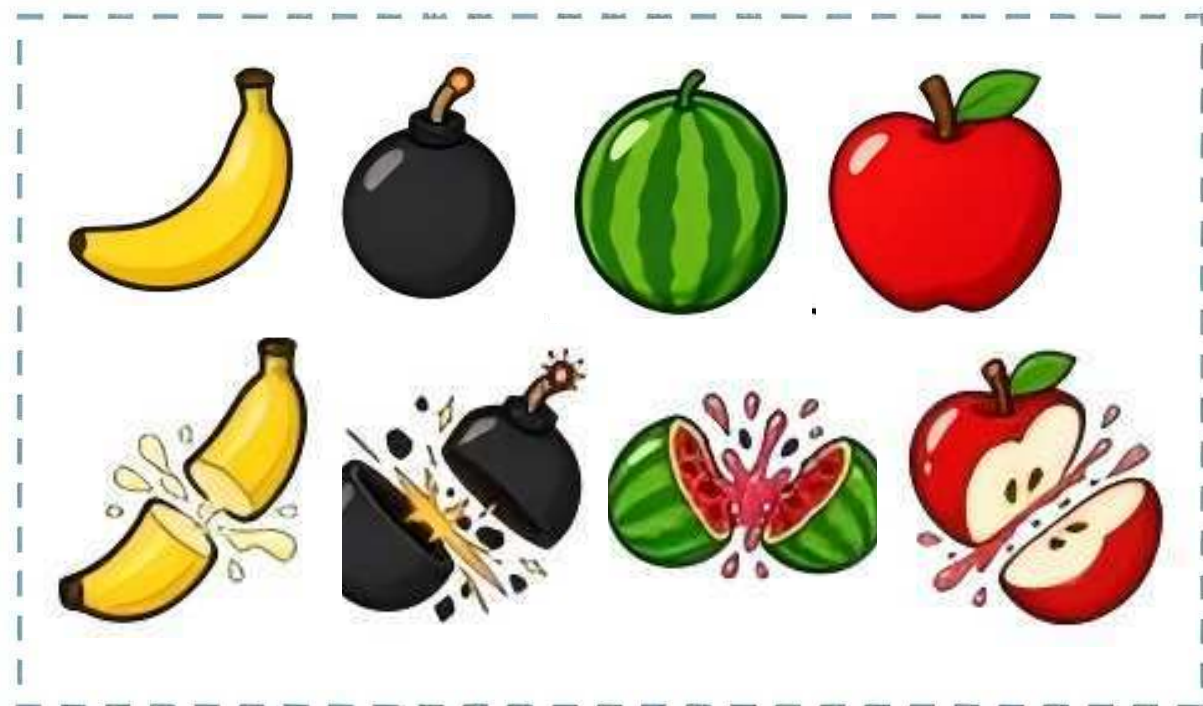
### Parallel Execution

3 independent hardware instances compute matrix rotation for 3 game slots simultaneously at 50 MHz.

# ROM Storage Strategy: Maximizing M10K Efficiency

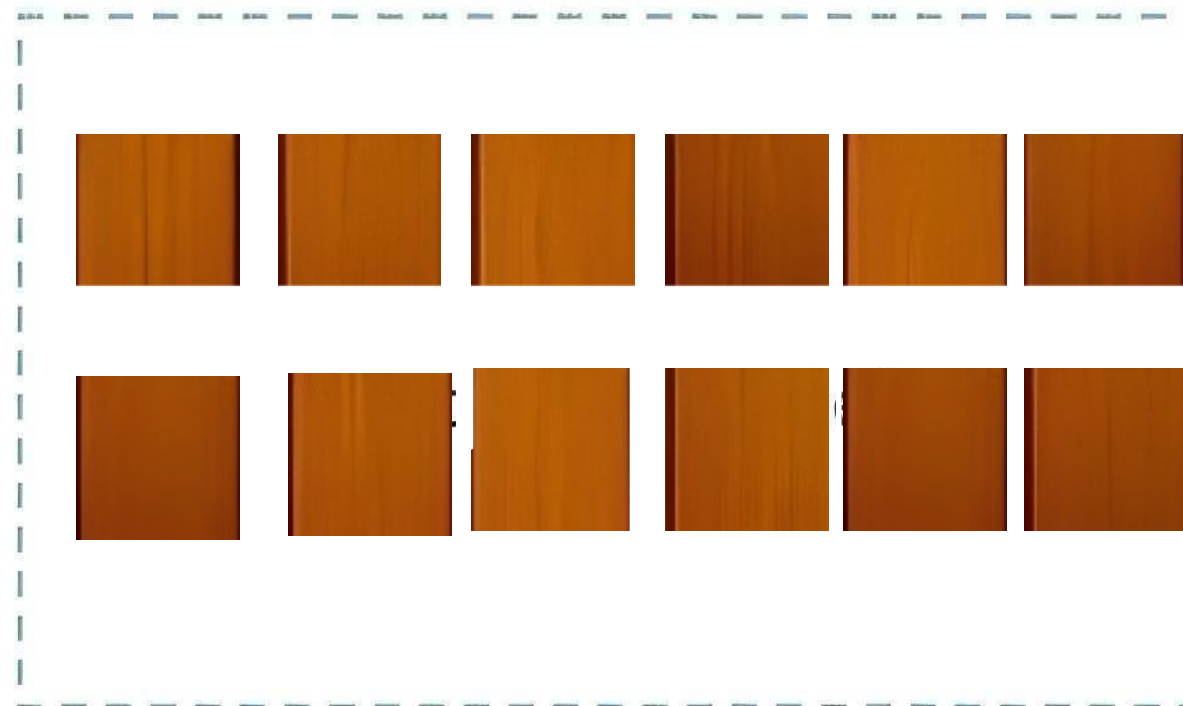
## Dynamic Sprites

- 64x64 RGBA4444 format
- 24 total ROMs (3 slots x 8 textures)
- Hardware transparency (A=0 is transparent)

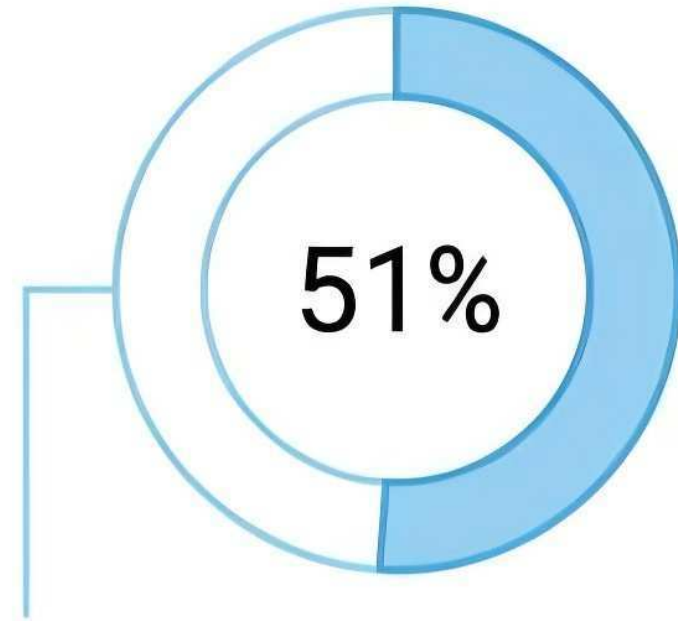


## Background Tiles

- 32x32 RGB888 format
- 12 ROMs arranged in 4x3 grid, repeating to fill 640x480

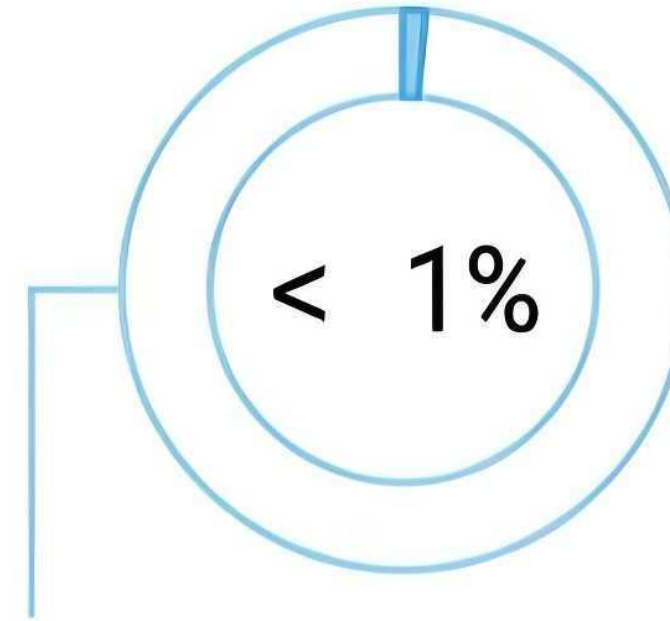


# Architectural Efficiency & Resource Utilization



M10K Memory Blocks

~204 out of 397 blocks used. 168 blocks dedicated to 24 Sprite ROMs; 36 blocks for 12 Tile ROMs.



Logic Registers

~400 FFs out of ~85,000 LEs utilized.

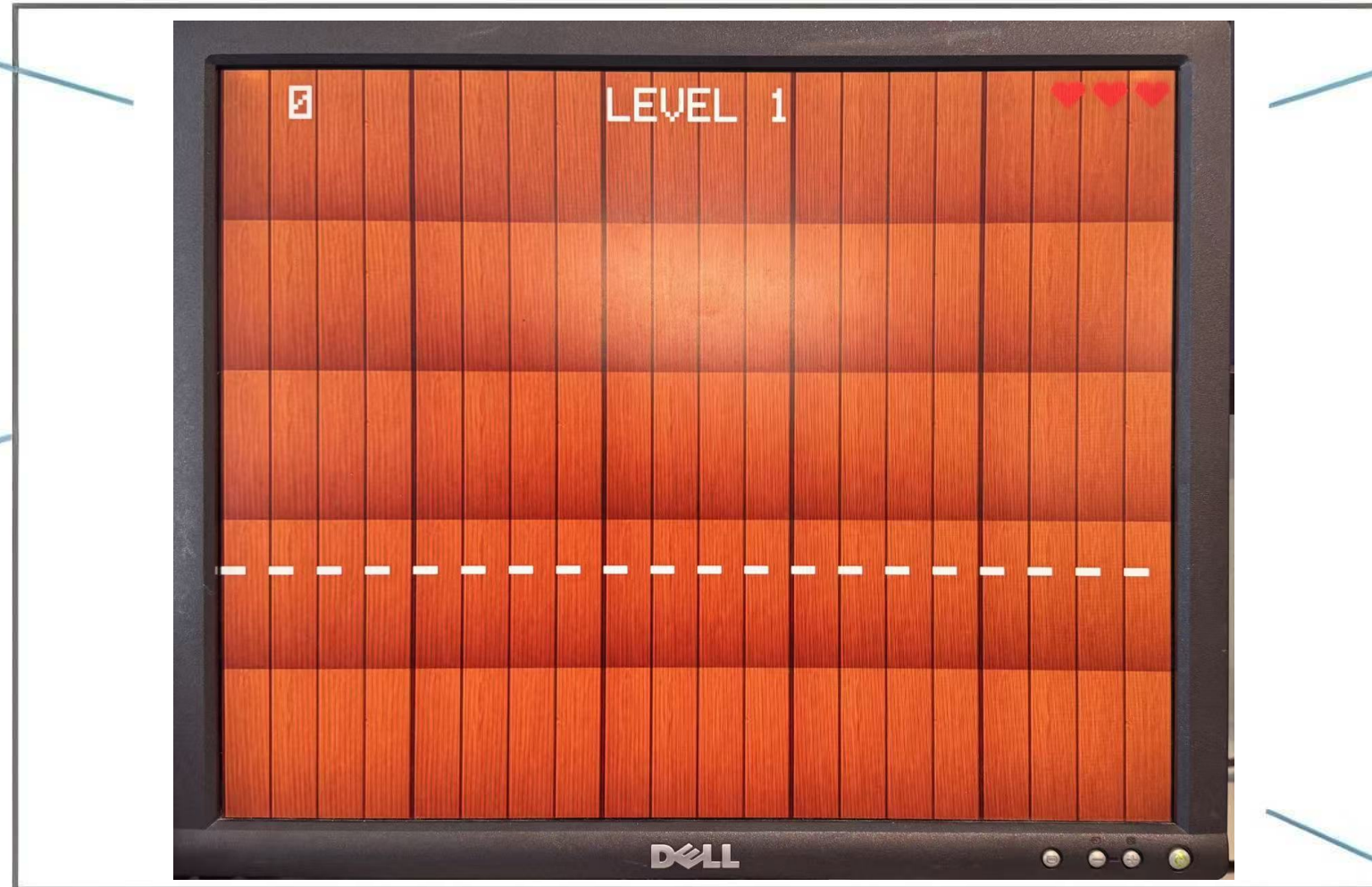
**By eliminating the SDRAM frame buffer and computing pixels synchronously at 50 MHz, the system outputs rich 640x480 VGA graphics at 60Hz with virtually zero bus bandwidth latency.**

# VGA HUD Composition

Score & Level Overlay  
(5x7 bitmap font)

Life Hearts  
(Red = active,  
Gray = lost)

Dashed Cut Line



Repeating  
Background Tiles

# The Hardware/Software Interface: Avalon-MM Register Map

**Figure 2 — Avalon-MM Register Map**

Base = 0xFF200000 | addressUnits = WORDS | readWaitTime = 1, writeWaitTime = 0

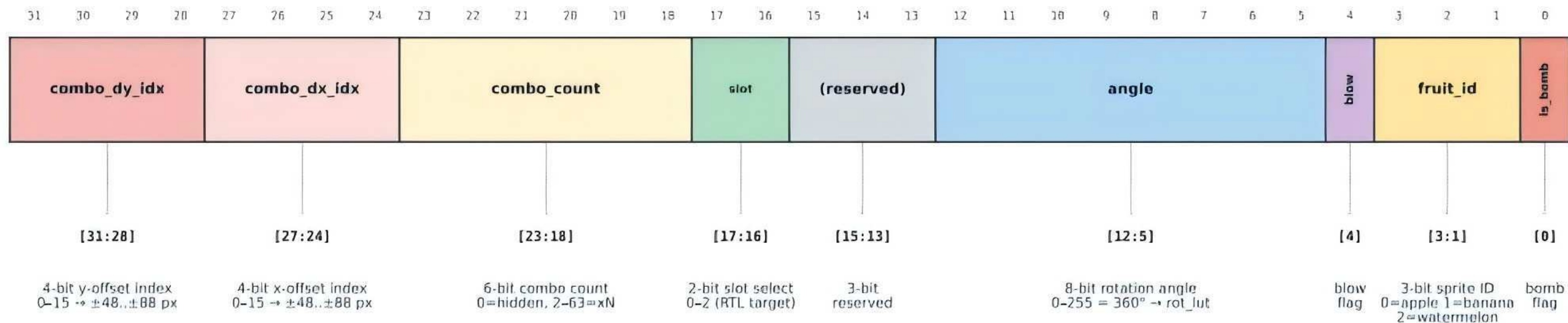
Word Addr	Byte Offset	Name	Access	Width	Description
0	0x00	ULTRA_STATUS	R/W1C	32	bit0=new_sample(W1C), bit1=echo, bit2=trig, [31:3]=0
1	0x04	LAST_ECHO_CNT	R	32	Last echo pulse width in 50 MHz ticks; cm = val/2900
2	0x08	CURR_ECHO_CNT	R	32	Current (in-progress) echo count; debug only
3	0x0C	SAMPLE_ID	R	32	increments on each measurement; used for edge detect
4	0x10	FRUIT_X	W	10	X position of selected slot (0-639)
5	0x14	FRUIT_Y	W	10	Y position of selected slot (0-479)
6	0x18	FRUIT_RADIUS	W	10	Radius of selected slot (typ. 22-34)
7	0x1C	FRUIT_CTRL	W	1	bit0 = visible for selected slot
8	0x20	SCORE	W	32	Current score; RTL derives LEVEL X display from this
9	0x24	GAME_STATE	W	Complex	Complex bit-field: bomb/sprite/blow/angle/slot/combo (see detail below)
10	0x28	LIVES	W	2	Lives remaining (0-3); drives heart display
11	0x2C	OBJECT_INDEX	W	2	Select target slot (0-2) for regs 4-7

Read-only (from ultra\_sensor\_csr)
  Write-only (HPS → FPGA)
  R/W (W1C clear)

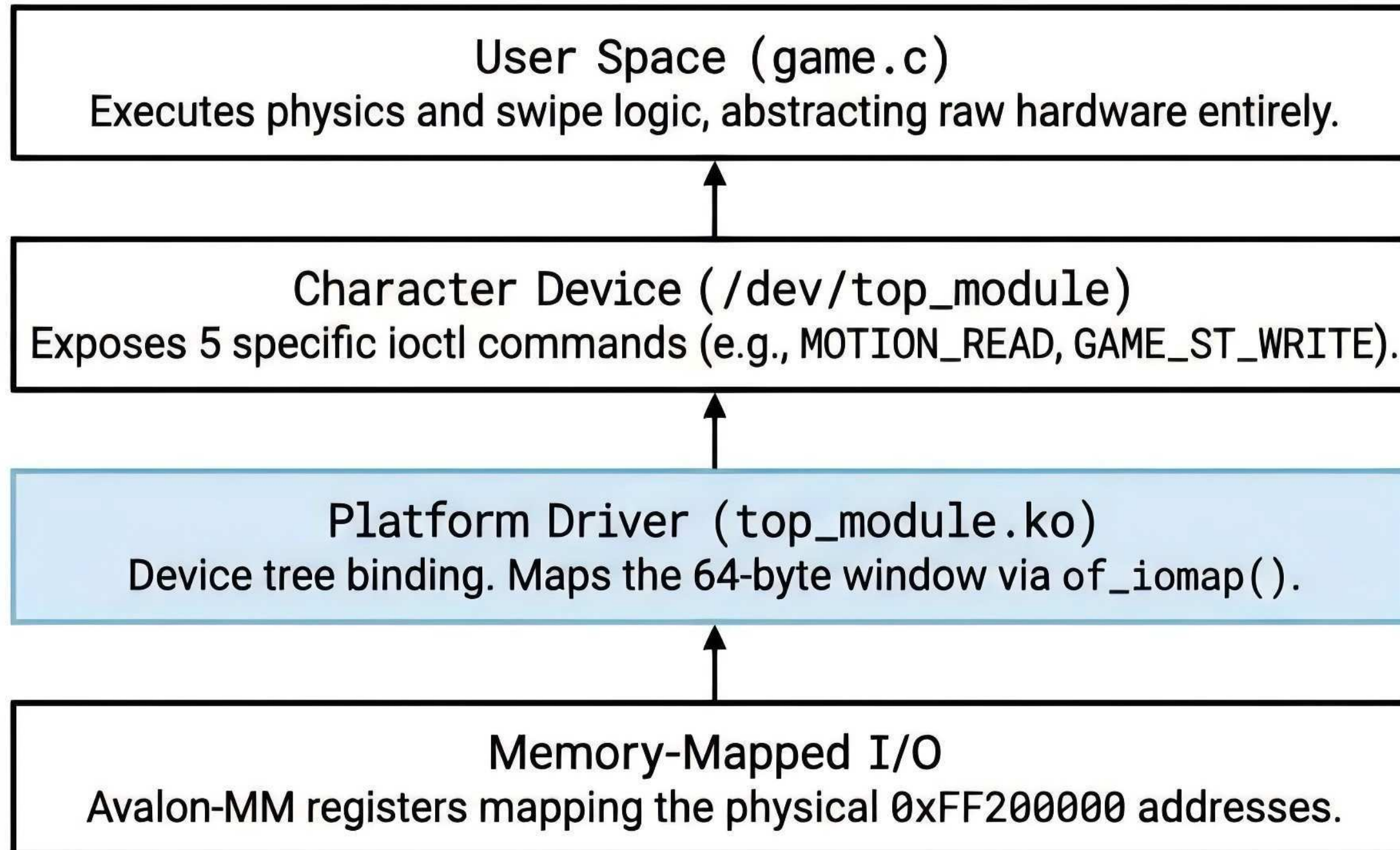
# Deconstructing GAME\_STATE (Offset 0x24)

## GAME\_STATE Register (0x24) — Bit-Field Detail

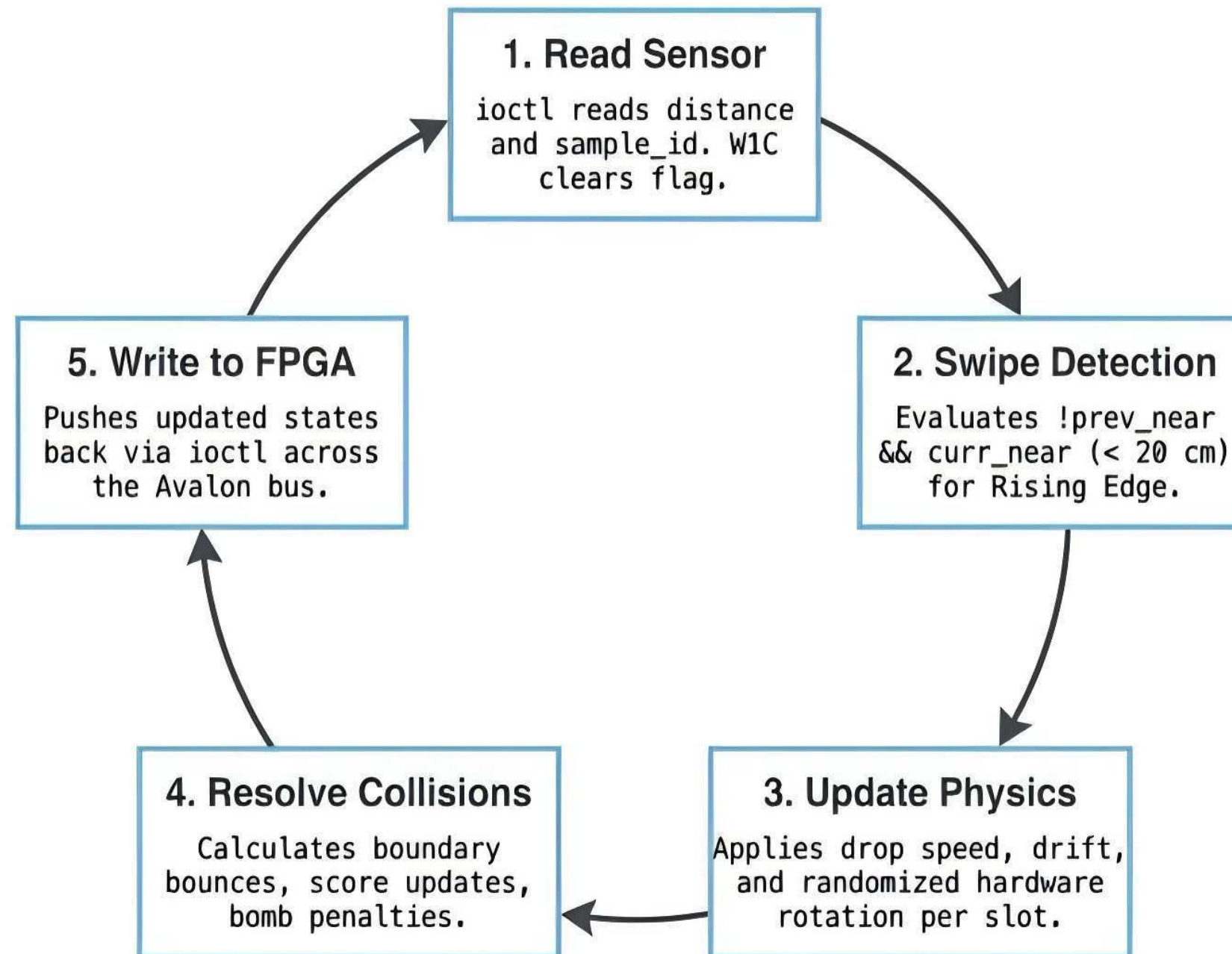
Written per-slot; bits[17:16] select the target game\_state\_reg[] (not OBJECT\_INDEX)



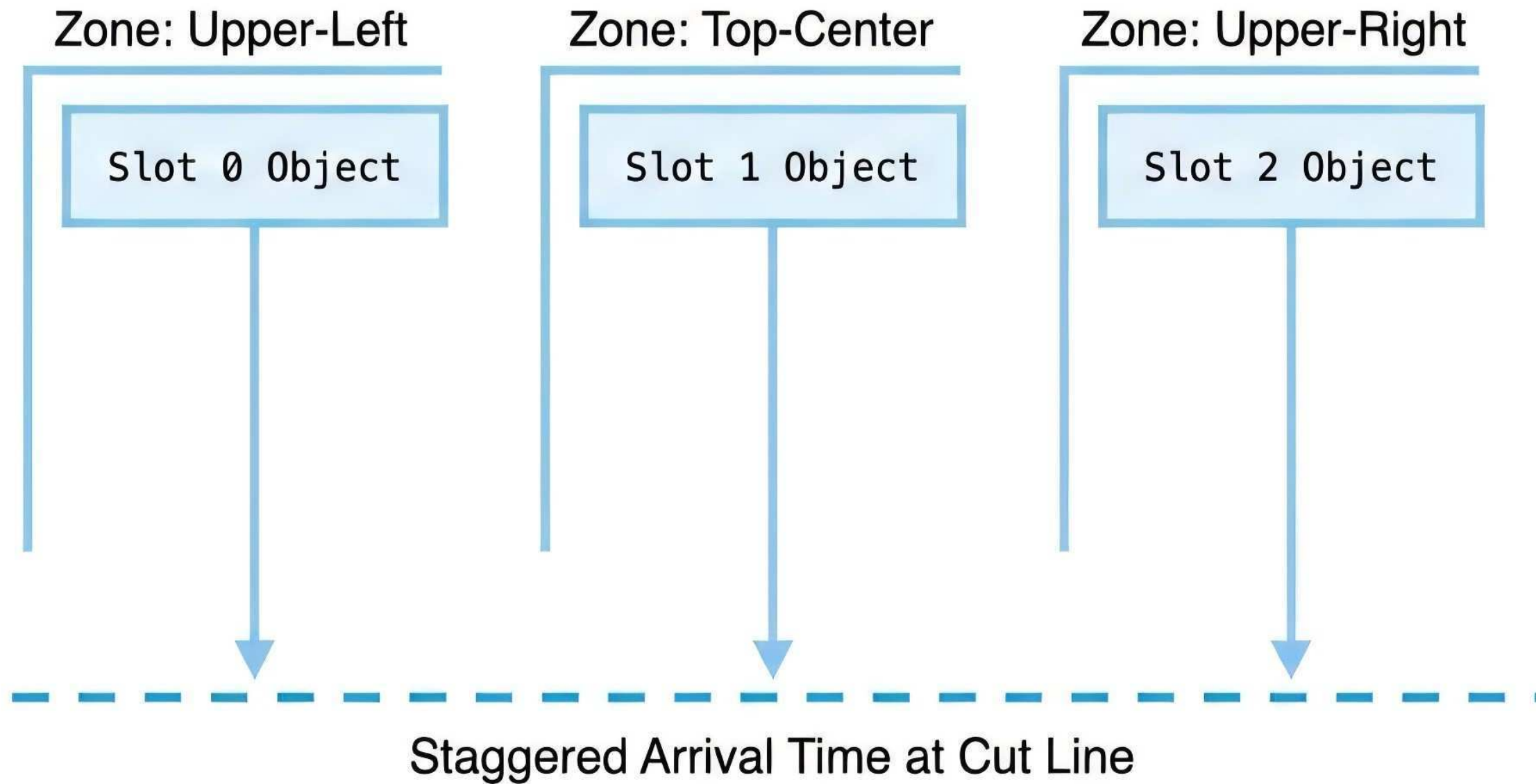
# The Software Stack: Kernel to User Space



# Real-Time Game Logic: The 30 FPS Loop



# Spawning & Physics Engine



## Metrics Generator:

- 70% Fruit / 30% Bomb probability
- 3 random rotation speeds (Fast, Med, Slow)
- Variable horizontal drift (VX\_MAX\_Q)

# System Demonstration

DEMO PART

Watch For:

- **Level Scaling:** 1 to 3 concurrent objects.
- **Hardware Rotation:** Smooth sub-pixel rotation.
- **Combo System:** Real-time HUD overlays.
- **Edge Detection:** Rejection of static hands.

# Questions?

---

Team 3: Peiheng Li, Chengrui Li, Yitong Bai  
CSEE 4840 Embedded Systems Design