Hardware Accelerated Super Resolution & Framerate Upscaling

ECE532 Design Project
Group 2

Yong Da Li, Benjamin Cheng, Jay Mohile, Leo Han

Rubric: https://piazza.com/class/lcia1n1v3jyfz/post/64

Intro: 2 min

title of project and team members: **use pre-existing (title)**

overview of the topic to remind us what you are doing: **use preexisting (solution 1 / 2)**

Review: 3 min

proposed system from last time - block diagram: 1 min

current system block diagram - what is different and why

- Simplified processing: found a way around pre-fetching: 1 min
- More Complex DMA: 1 memory/processing (needs Microblaze involvement and synchronization)

challenges so far: 1 min

- HDMI is hard → need to use many pre-packaged IPs and correctly configure signals/registers, sometimes doesn't pass timing in implementation

what did you demo?

- Ben - HDMI in
- Yong Da - HDMI out
- Jay - frame interpolation and upscaling (Mario) -> show precanned images
- Leo - dma

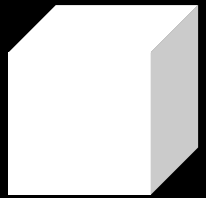
what's left to do? - any changes to fit the remaining time: 1 min

- HDMI pass-through via RAM
- Merge processing and HDMI with RAM → needs some kinda synchronization (ex. "New frame valid" and "HDMI output ready"?)
- Basically:
  - Link In to Out
  - Link In to Processing

what do you plan to do for the final demo?

Solution

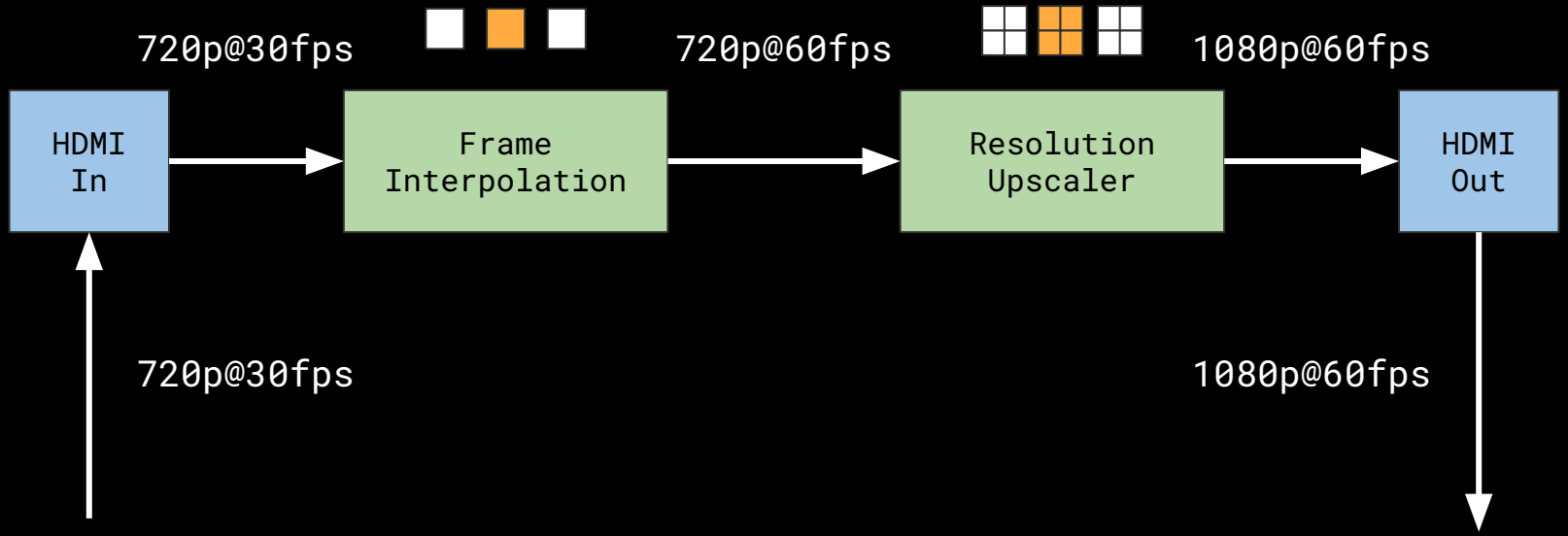
HD@30FPS



up-scaler

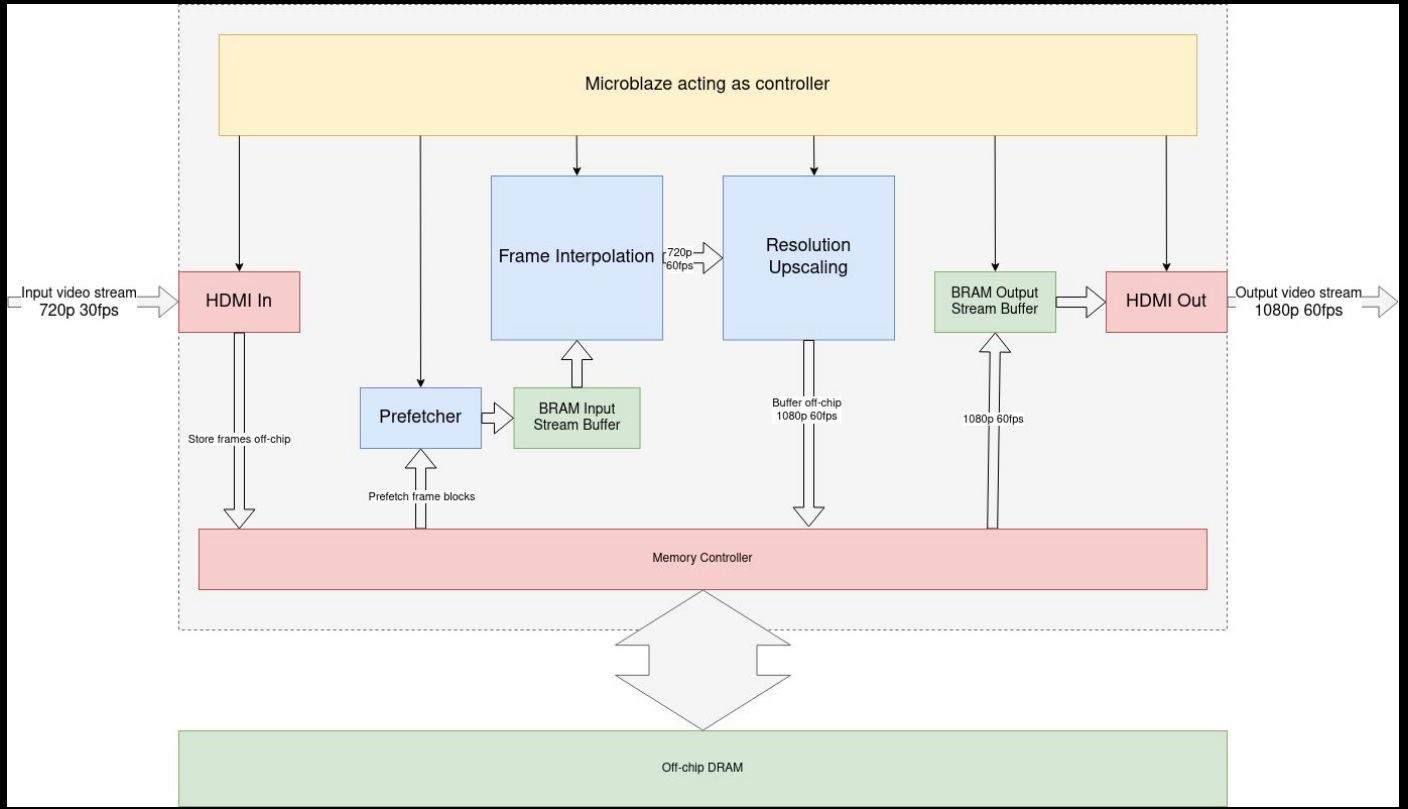
FHD@60FPS

Solution

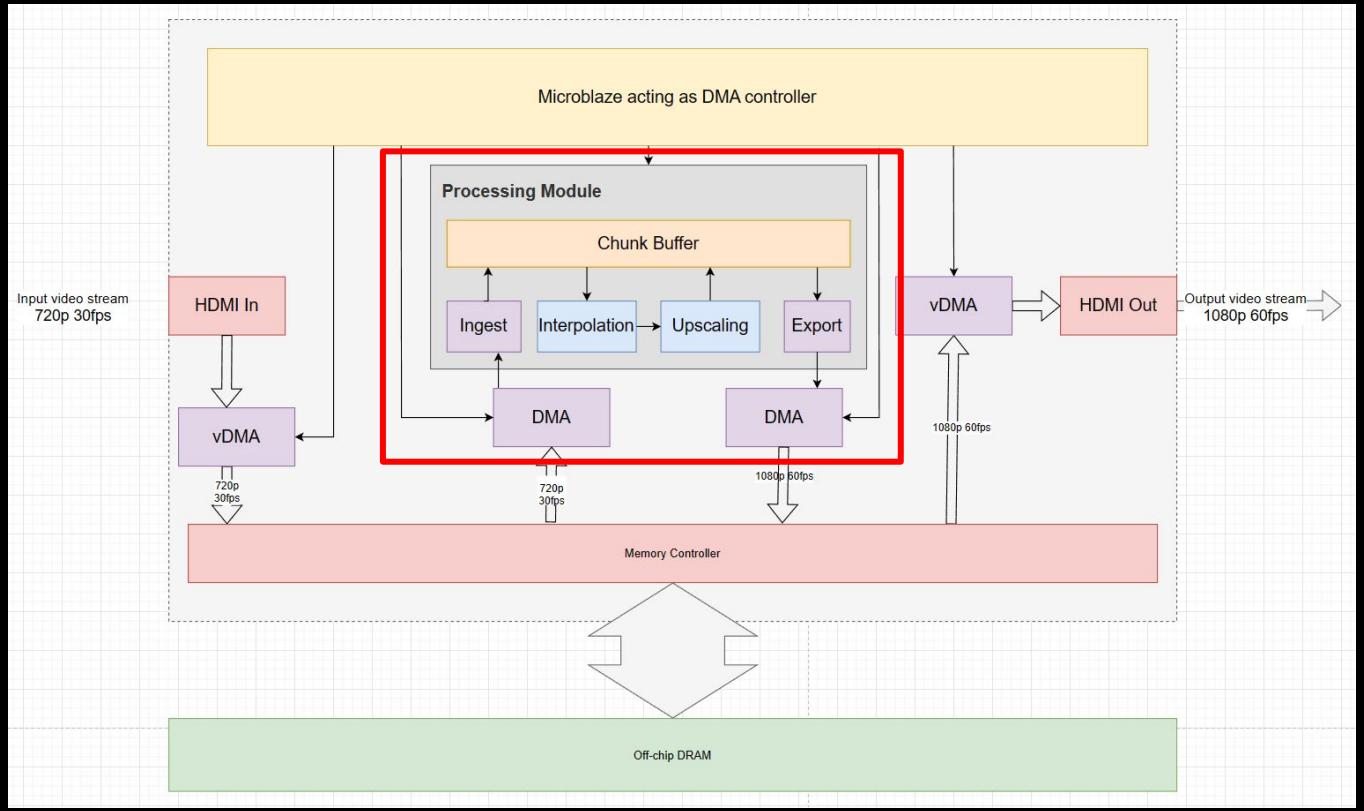

720p@30fps
720p@60fps
1080p@60fps
HDMI In
Frame Interpolation
Resolution Upscaler
HDMI Out
720p@30fps
1080p@60fps

Changes from last time

Previous: System Architecture



Current: System Architecture



Challenges so far

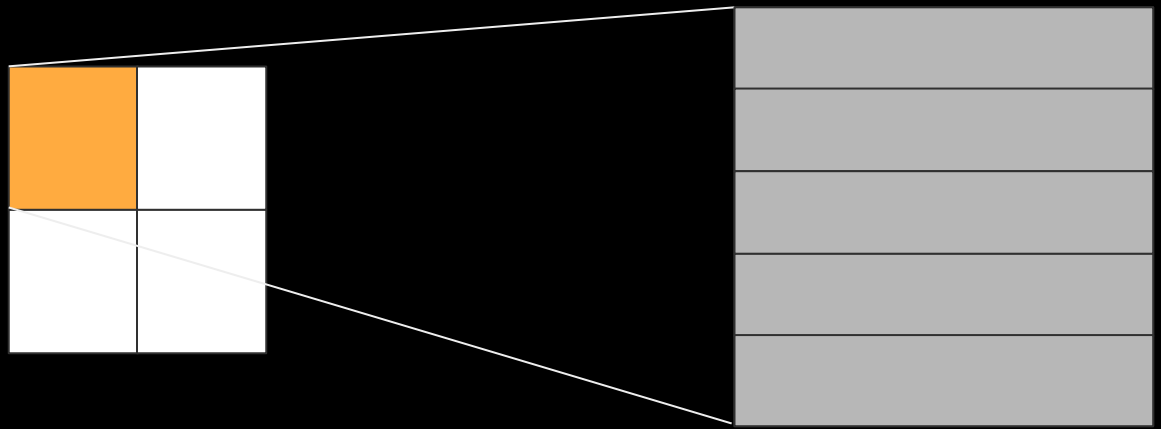
HDMI is hard

- Need to use many IP blocks with correct configuration
  - Struggled with correct clocks and enable tie-up/downs, DDR3 caching
- Input: Rbg2dvi decoder, AXI4-stream to video-in, AXI video DMA
- Output: same as input in reverse + video timing controller

- Took much longer than our initially planned 1-2 weeks

DMA Scatter/Gather

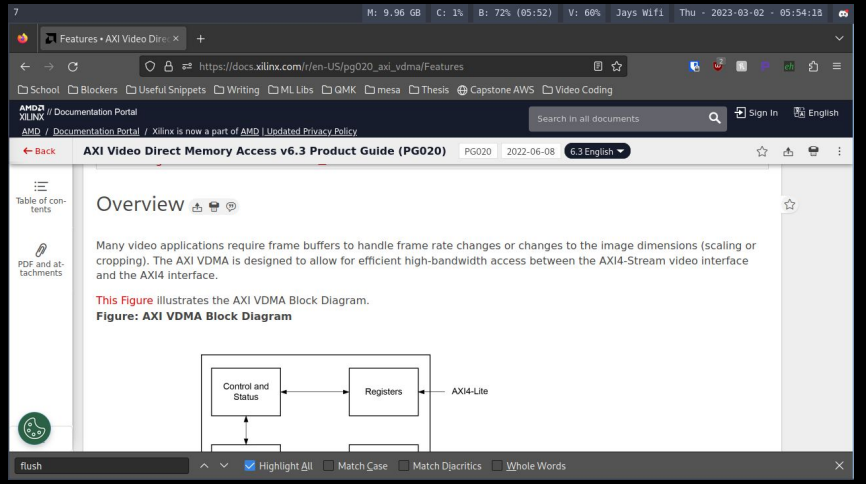
Processing tiles in 2D Frames → **Discontiguous memory address**



What we demo’ed

HDMI Input

- Successfully capture HDMI Input frames into specified location in DDR3
- Framebuffer read out on Microblaze, and dumped through UART
- Dump converted into PNG through Python

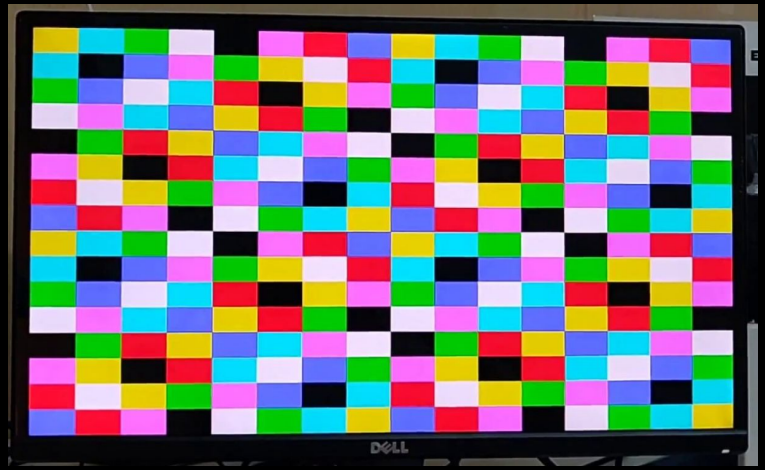


- Evidence of “live” capture seen through changing timestamp

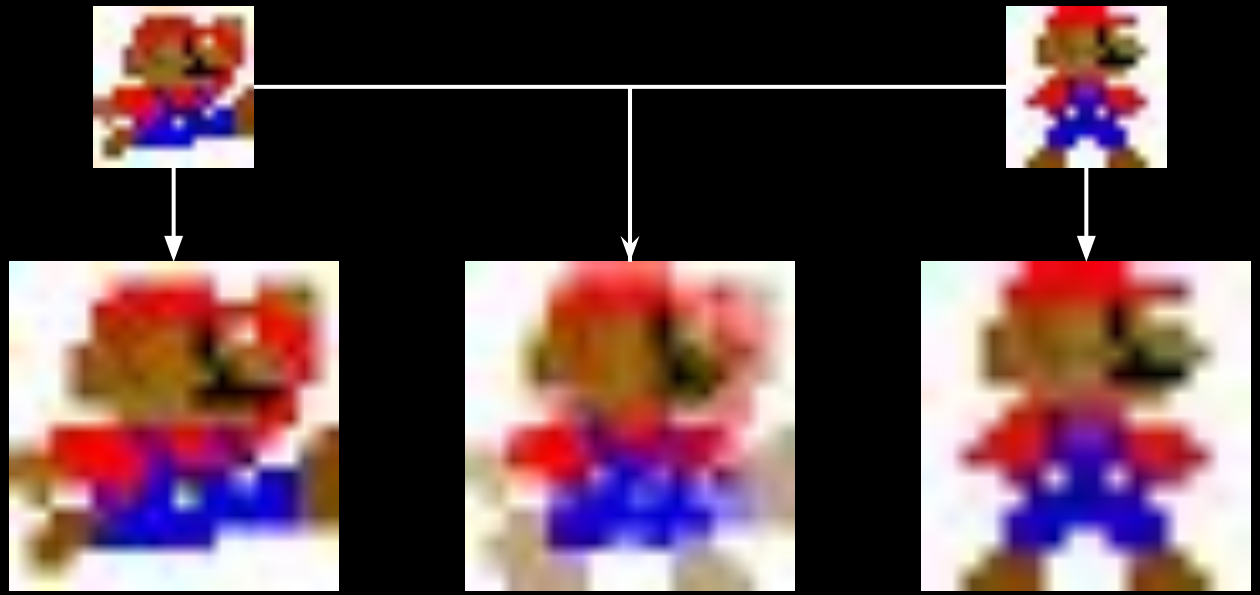
HDMI Output

- Video test pattern generator
- Pass-through (on 2016.4)

- Not working: stream from RAM

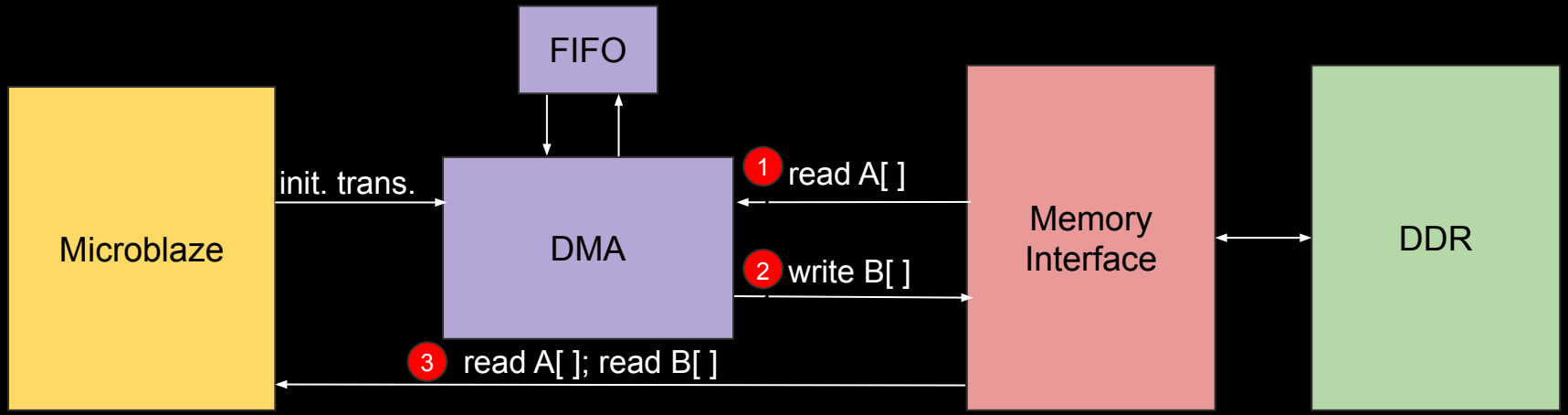


Frame Interpolation and Upscaling



DMA

- Demonstrated DMA with DDR using standard Xilinx DMA IP blocks controlled by Microblaze
- Copied data from one address in DDR to another via DMA
- Verified correct data copy by reading back data from DDR via AXI-4



What’s left to do?

Integration

- Stream video from memory-mapped DRAM address
- Add AXI-S master output to processing block for interfacing with DMA IP
- Implement synchronization signals
  - Microblaze interrupts
    - Video-in to Microblaze to signal portion of frame is ready (i.e. written to memory)
    - DMA to Microblaze to signal transaction complete
- Integrate memory buffering (DMA) with processing and video in/out

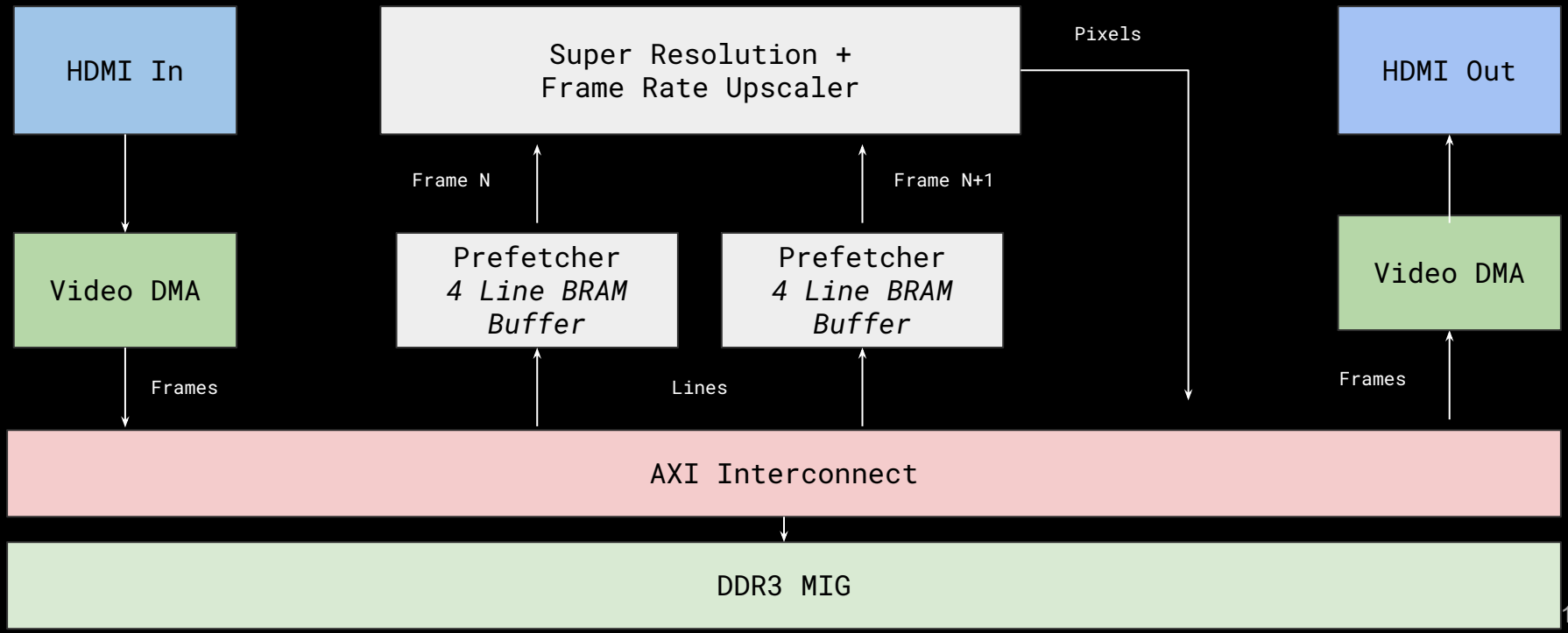
Final Demo Plans

Laptop playing 720p 30FPS video split into two output HDMI video stream.

The first stream goes to FPGA which upscales and doubles frame for an output HDMI video stream at 1080p 60FPS going to demo monitor.

The second 720p 30FPS stream goes directly to another demo monitor.

Proposed System Architecture

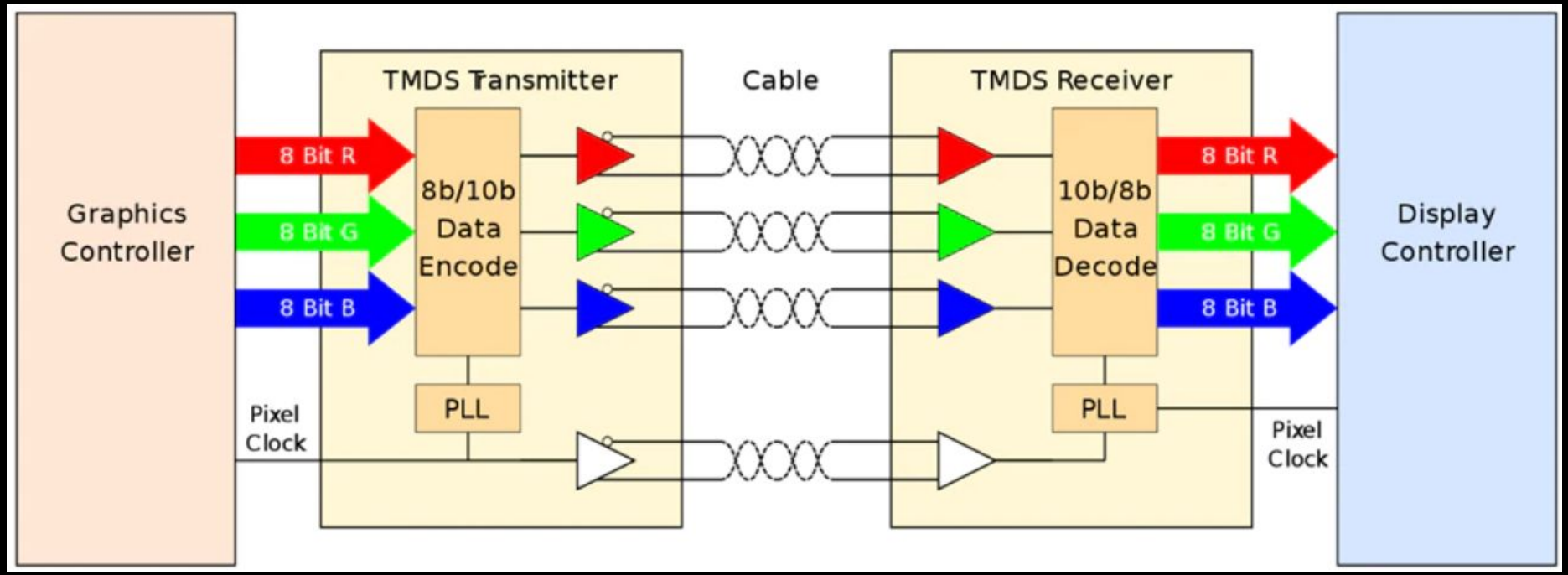


Motivation

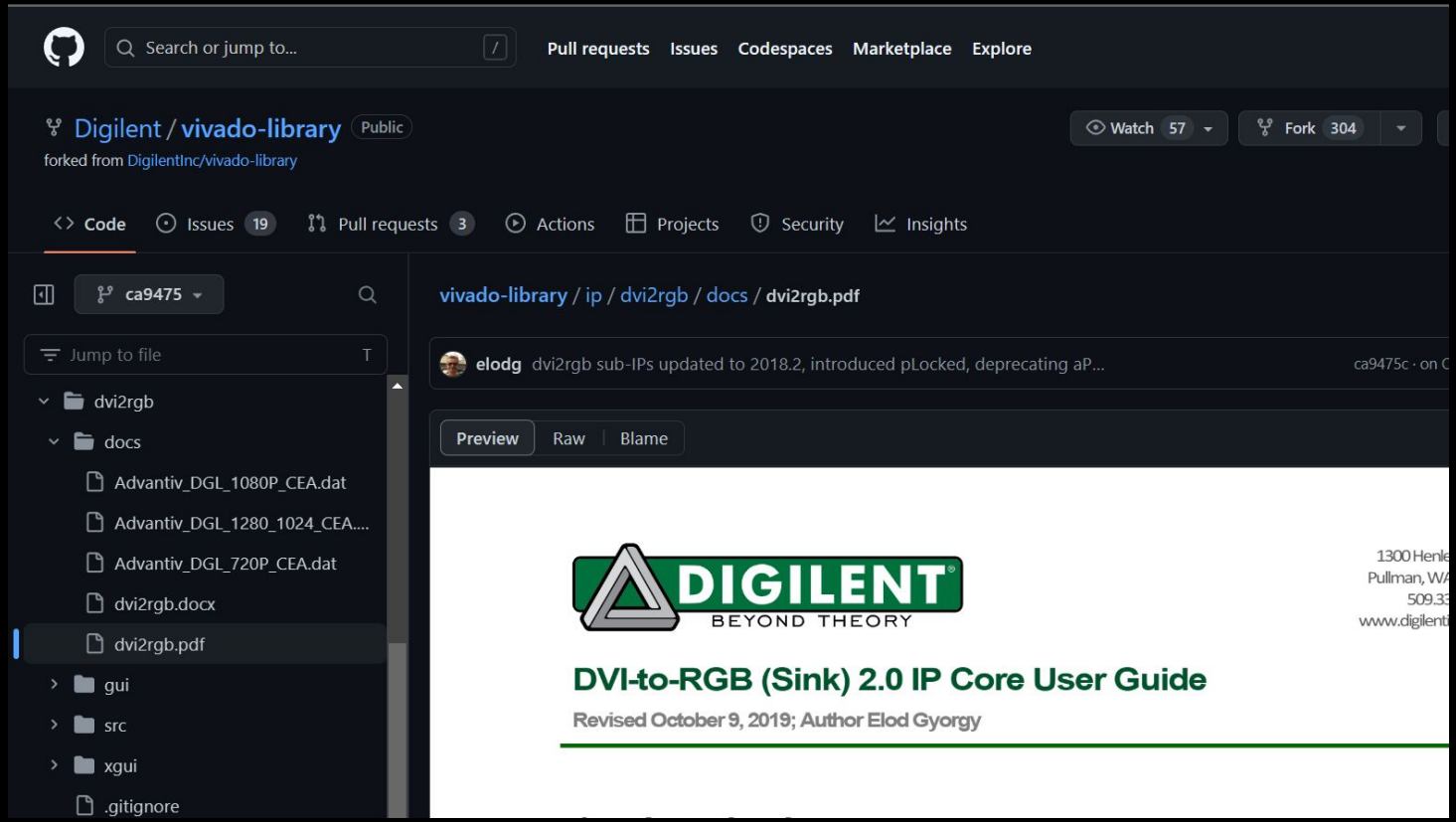
1. Real-time rendering 3D content (e.g. gaming) is compute intensive and hardware (GPUs) is expensive.
2. Video sources may be low quality (low resolution, low frame rate)

Component Architectures

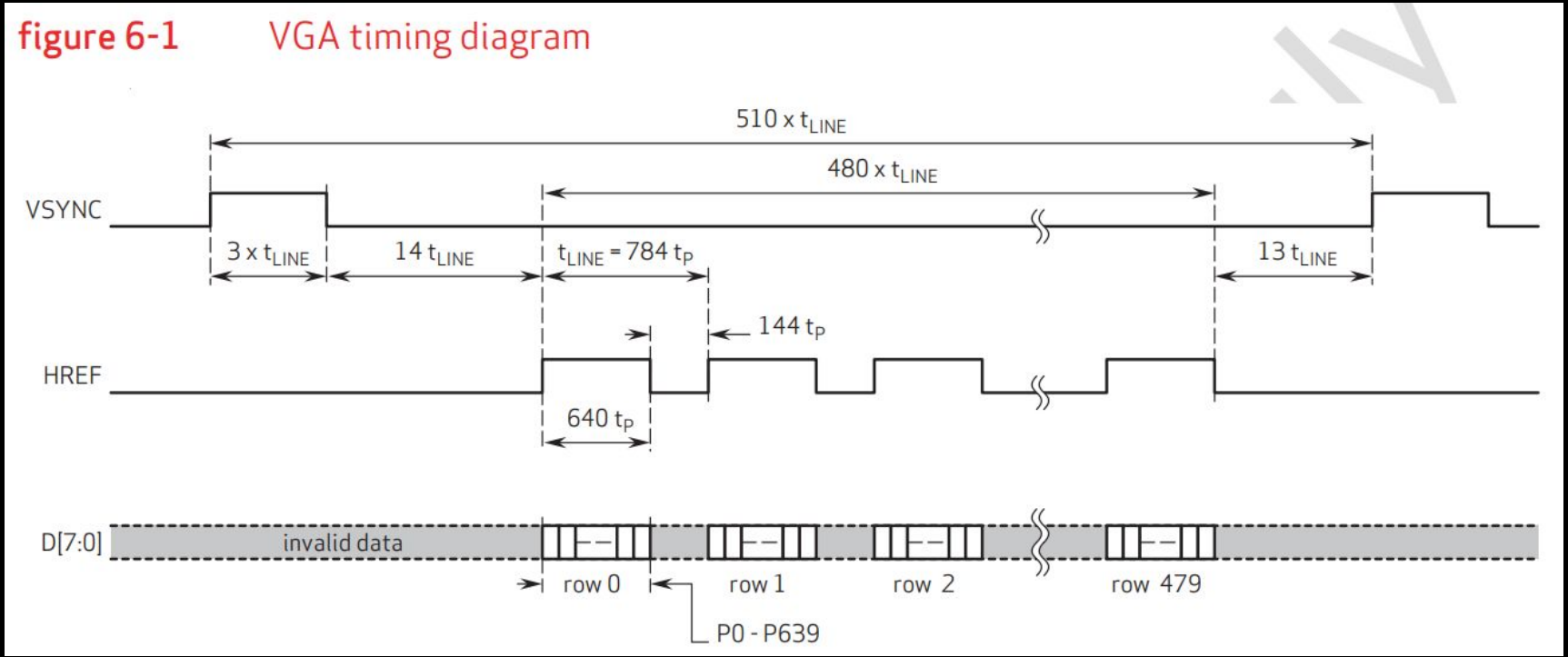
HDMI Video Processing



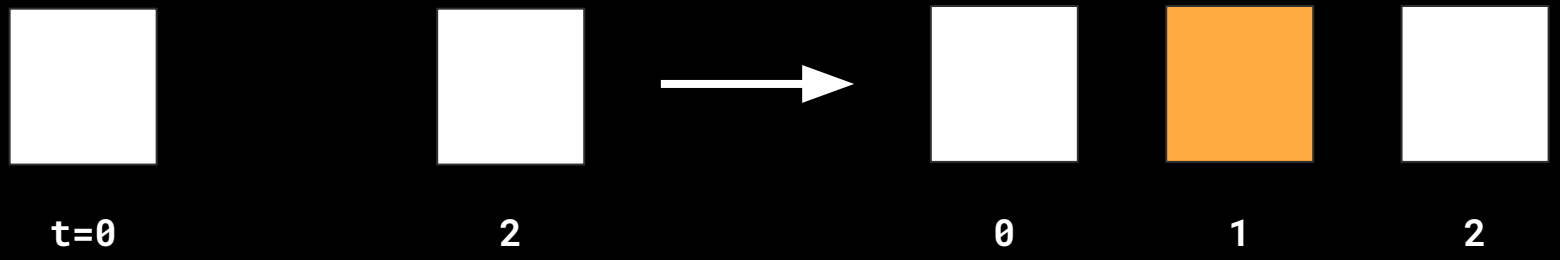
HDMI Video Processing



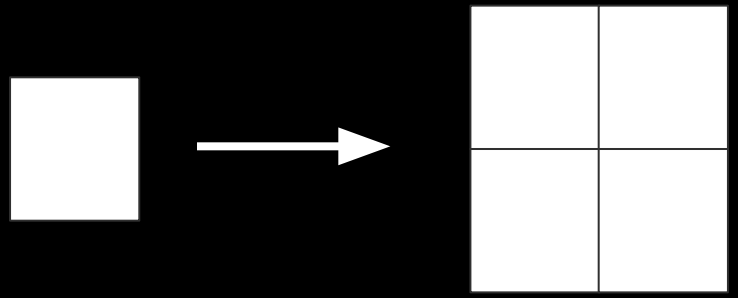
HDMI Video Processing



Framerate Upscaling

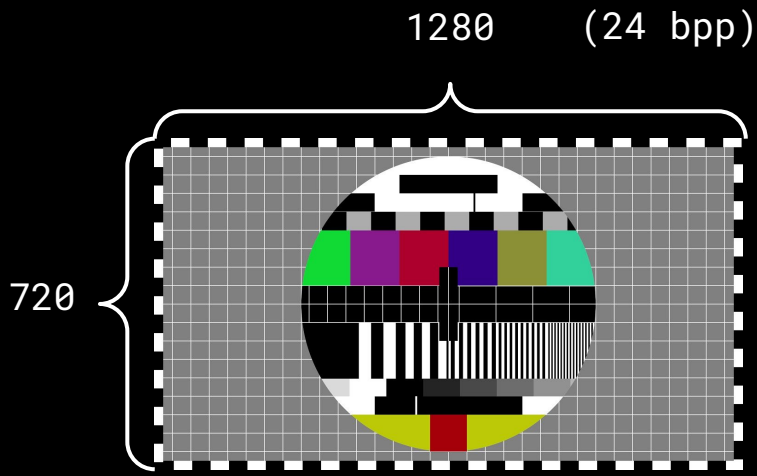


Resolution Upscaling



720p ***1080p***

Key Challenges

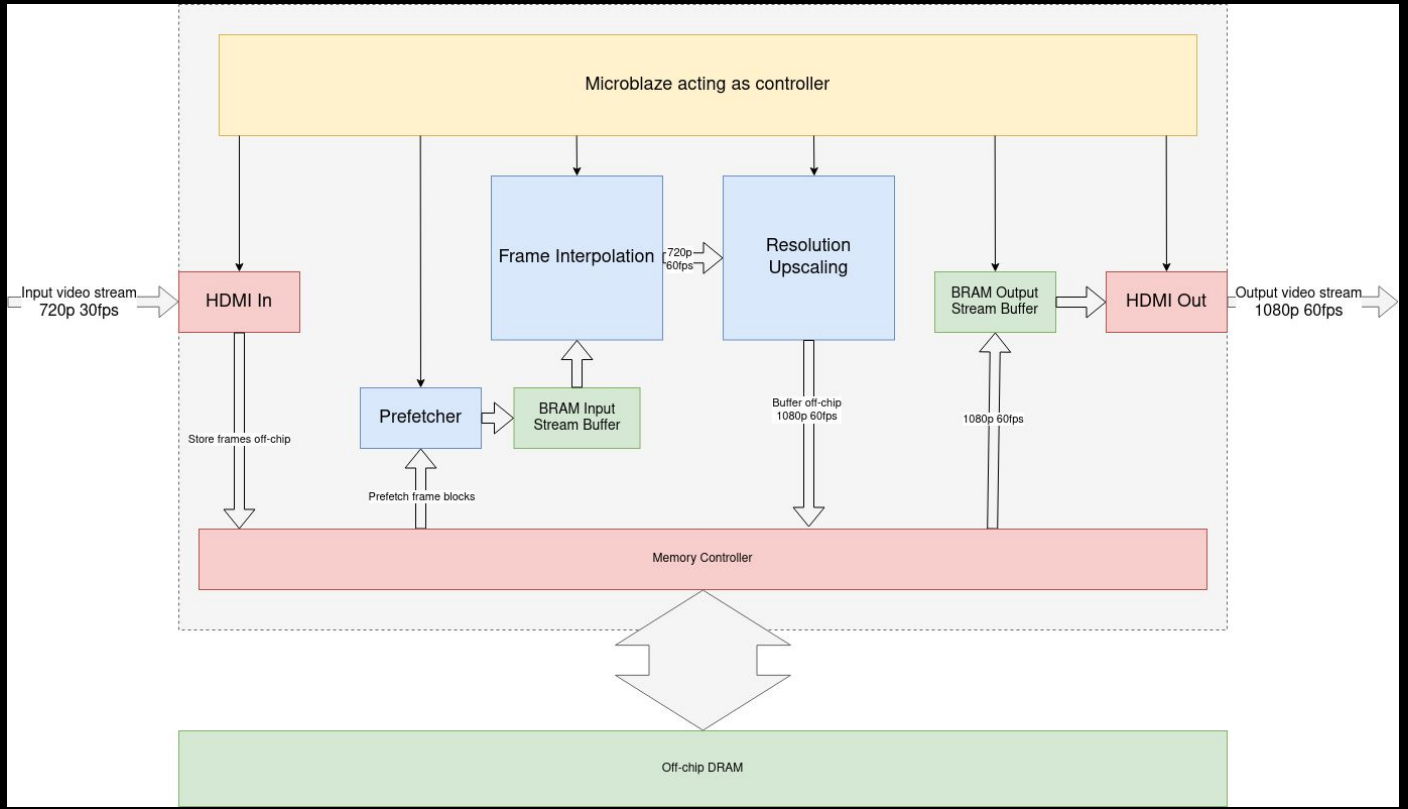


> 22 Mbits

>

13,140 kbits
BRAM

System Architecture



Milestones

Video Upscaling Interpolation Control

Milestone 1

Design + Environment

Video Upscaling Interpolation Control

Milestone 2

HDMI

Video Upscaling Interpolation Control

Milestone 3

Passthrough

Video Upscaling Interpolation Control

Milestone 4-5

Algorithms

Video Upscaling Interpolation Control

Milestone 6

Control

Video Upscaling Interpolation Control

Q&A