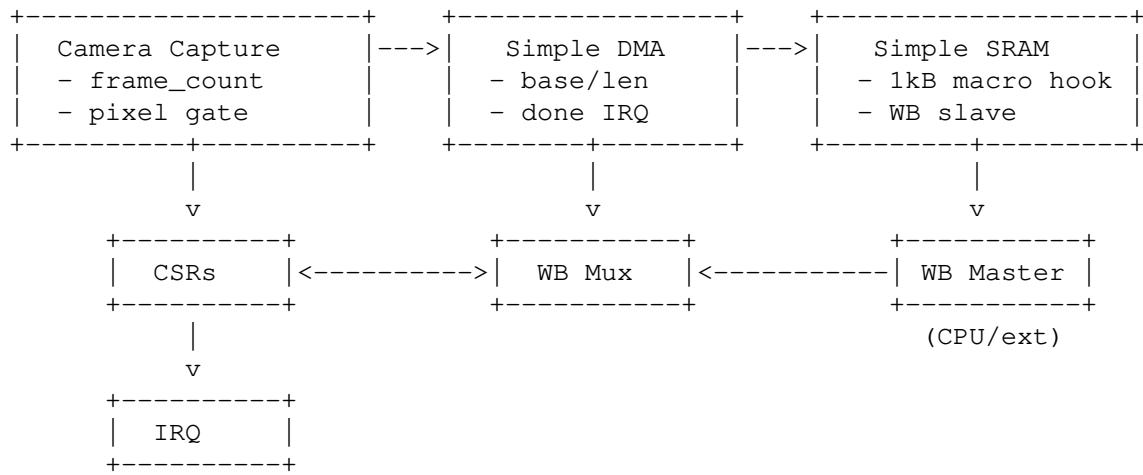


Dashcam SoC Micro-Architecture Specification (MAS)

1. Overview

This document describes the internal micro-architecture of the Dashcam SoC mock design. The design is optimized for deterministic simulation and pre-silicon flow validation.

2. Micro-Architecture Block Diagram



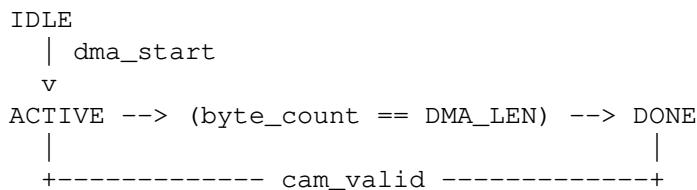
3. Camera Capture

- Inputs: `cam_valid`, `cam_sof`, `cam_pixel[7:0]`.
- Outputs: gated pixel stream to DMA, frame counter.
- Behavior:
 - `frame_count` increments on `cam_sof`.
 - Pixels are forwarded to DMA when `CTRL.cam_en==1`.

4. DMA Engine

- Simple linear write to SRAM using `DMA_BASE` and `DMA_LEN`.
- Handshake:
 - `CTRL.dma_start` triggers a transfer.
 - On completion, `dma_done` asserted and `IRQ` pending set.

DMA State Flow



5. SRAM Subsystem

- Primary: `sky130_sram_1kbyte_1rw1r_32x256_8` (macro hook).
- Simulation: behavioral SRAM in RTL.
- Interface: Wishbone-style read/write for CPU/external master.

6. CSR Block

- Generated from SystemRDL (`ips/csr_regs/dashcam_csr.rdl`).
- Registers include control, DMA parameters, IRQ status/clear, IOMUX, build info.

7. Interrupt Controller

- Single interrupt source from DMA completion.
- `IRQ_CLEAR` pulse clears pending state.

8. Reset / CDC / RDC

- Single clock domain.
- Reset is async assert, sync deassert via `reset_sync`.
- CDC checks are clean by construction (single clock). RDC satisfied by reset synchronizer.

9. Timing Targets

- 200MHz @ 1.8V (Sky130 FD-SC-HD).
- SDC: `constraints/sky130/dashcam_soc_top.sdc`.

10. Integration Notes

- Top module: `top/dashcam_top/rtl/dashcam_soc_top.sv`.
- Wishbone map: CSR @ `0x1000_0000`, SRAM @ `0x2000_0000`.