AXI4-Lite Manifold: Canonical Base-Address Map

1,116 words · ~5 min read

---

*VCA-CSA-101 cross-chapter quick-reference handout. Anchors: §25 (canonical address map + 3-pattern overview); §5.7.1 (Virtus Console memory map); §12.6 (Driver-Layer OS Services). Companion to `cross-chapter-doorbell-scratchpad-descriptor.md` (interaction patterns) and `cross-chapter-stdlib-service-reference.md` (OS-stdlib contract). *

Purpose: the single-page reference card for every AXI4-Lite slave window in the Virtus Console's IP Pack manifold. Print and pin during Lab 5.4 (Virtus Console bring-up), Lab 6a.3 (linker bring-up; la-pseudo resolves MMIO base constants), Lab 11.3 (HDMI end-to-end), Lab 12.4 (Console drawing capstone), Lab 12.5 (audio-capstone silicon-cert). Drift between this map, the IP Pack HDL register addresses, and student programs touching MMIO is a curriculum bug, the audit cycle catches and reconciles such drift.

At a glance

Property	Value
Routing rule	`addr[31] == 1` → MMIO manifold; `addr[31] == 0` → BRAM (instr_mem + data_mem)
Slave-select decode	bits `[19:16]` of the 32-bit AXI address
Slave-window stride	64 KiB (`0x10000`). Peripheral N at base `0x801N0000`
Per-slave window size	4 KiB nominal (HDMI is 64 KiB to accommodate the tile-map RAM)
Handshake protocol	AXI4-Lite, 5-signal (AWVALID/AWREADY/WVALID/WREADY/BVALID); combinational AWREADY canonical for new slaves
Misaligned-access behaviour	IP-Pack convention: discard write / read zero (no bus-error trap in CSA-101; CSA-201 introduces PMP)
Unmapped-region behaviour	Same as misaligned: write discarded / read returns zero
Source-of-truth (HDL)	`peripheral-ip-pack/hdl/manifold/axi_manifold.v` (M0-2 R5.1 commit `41d5df3`)
Source-of-truth (software headers)	`peripheral-ip-pack/sw/{hdmi,audio,ps2,gpio}_demo/*.h`

§1: Canonical address map

Base address	Window	Peripheral	Doorbell	Scratchpad	Descriptor-ring
`0x80000000`	64 KiB	**Screen**. Pixel framebuffer (320×240 12bpp post-R6B.2 BRAM uplift)	(none; CPU writes directly)	`0x80000000..0x8000FFFF`	(n/a)
`0x80030000`	16 B	**Sys**. System control (cycle counter at `+0x00`; trap-vector at `+0x04`; halt at `+0x0C`)	(none)	(4 register slots)	(n/a)
`0x80100000`	64 KiB	**HDMI / Console**, 80×30 tile-map; ASCII + 16-color palette	vsync `+0x20`	tile-map `+0x000..+0x95F`; palette `+0x100..+0x13F`	frame-list `+0x200..+0x2FF` (M0.5 forward)
`0x80110000`	4 KiB	**Audio**, PWM-RC at 22 kHz; 8-bit mono	half-empty `+0x10`	sample-buffer `+0x100..+0x1FF`	playback queue `+0x100..+0x1FF` (head `+0x200`, tail `+0x204`)
`0x80120000`	4 KiB	**PS/2 keyboard**. Clean-room AT/PS-2 protocol decoder; shift-only modifier MVP	scancode-ready `+0x08`	scancode FIFO `+0x10..+0x1F`	(n/a)
`0x80130000`	4 KiB	**GPIO**, 16-bit bidirectional regfile; tristate-tuple pattern	INT_STATUS `+0x0C` (W1C)	DIR `+0x00`; OUT `+0x04`; IN `+0x08`	(n/a)
`0x80140000`	4 KiB	**DS2 / GamePad**, 4-wire SPI	(deferred to M0.5)	button bitmap `+0x00`	(n/a)

Base address	Window	Peripheral	Doorbell	Scratchpad	Descriptor-ring
		controller decoder; DS2 ⊃ SNES button set			
`0x80150000`	4 KiB	fpga_pio. Programmable-I/O state-machine substrate	(per-PIO doorbells `+0x80..+0xBF`)	program RAM `+0x100..+0x1FF`; FIFOs `+0x000..+0x07F`	(n/a)

Per-pattern column legend (cross-references `cross-chapter-doorbell-scratchpad-descriptor.md`):

- **Doorbell**. Single-bit MMIO offset; CPU writes-1 to ring; IP polls or interrupt-on-rising-edge.
- **Scratchpad**, CPU-readable IP-internal RAM; CPU writes via `sw`; IP reads internally on its own clock.
- **Descriptor-ring**. Circular buffer of fixed-size descriptor entries; CPU advances head; IP advances tail.

---

§2: Mnemonic guidance: the `0x801N0000` family rule

The Virtus Peripheral IP Pack uses a **64 KiB stride per slave window** beginning at base `0x80100000`. Peripheral N occupies the 4-KiB-nominal range starting at `0x801N0000`. The rule:

```
peripheral_base = 0x80100000 + (N * 0x10000)
```

where `N ∈ {0, 1, 2, 3, 4, 5}` for the M0-2 + M0.5 IP Pack roster. The 64 KiB stride leaves 60 KiB of unmapped padding above each 4 KiB slave window. A deliberate over-allocation that lets future slaves grow without renumbering existing ones.

Three industry conventions students will encounter, each with different design payloads:

1. **Linux device-tree-style allocation** (FreeBSD, NetBSD, Yocto-built embedded Linux): peripheral addresses encoded in a separate device-tree `.dts` file; kernel reads at boot; addresses are not constants in source. Pro: portable across SoC variants. Con: opaque if you only have the binary.
2. **Virtus convention** (this handout; CSA-101 baseline): peripheral addresses are constants in source code (compiler-emitted via the la-pseudo / linker prologue path). Pro: every address is grep-able from the program; the linker prologue resolves them at link-time. Con: not portable across hardware variants without recompile.
3. **Vendor-specific maps** (TI Sitara TRM, NXP i.MX RT TRM, Espressif ESP32 datasheet): peripheral addresses defined in vendor reference manuals; software headers (`*.h`) generated from the manuals. Pro: matches vendor toolchain expectations. Con: lock-in to one vendor's address-space layout.

Each convention solves the same problem (CPU and IP must agree where the IP lives) at a different level of indirection. CSA-201's driver-track introduces device-tree-style allocation as a forward-stretch lab; con-101 capstone may use vendor-style maps when shipping student-built peripherals to commercial silicon.

---

§3: Address-decode behaviour (boundary conditions)

The top-level `axi_manifold.v` module (M0-2 R5.1 commit `41d5df3`) implements address-decode as a 16-input multiplexer over the 5 currently-shipped slaves plus the framebuffer + system-control regions. Boundary cases:

- **Misaligned access** (e.g., `lw` at `0x80130001`): IP-Pack convention is to discard the write or return zero on read. RV32I-Lite does not raise alignment exceptions in CSA-101; CSA-201 introduces PMP-driven alignment-fault traps.
- **Unmapped region access** (e.g., `lw` at `0x80160000`, between `fpga_pio` and the next undefined slave): manifold returns zero for reads, accepts and discards writes. No trap. Students who write to an unmapped region see no effect; the silent-discard behaviour is itself a debugging cue.
- **Within-window unmapped offset** (e.g., `sw` at `0x80130020` when GPIO only defines registers up to `+0x0C`): per-slave HDL convention; most slaves treat as no-op. GPIO specifically returns zero on out-of-range reads.

- **Concurrent CPU + DMA access** (CSA-201 forward-pointer): CSA-101 has no DMA controller, so this case does not arise. CSA-201's manifold introduces an arbitration layer.

---

§4: Forward-compatibility notes (CSA-201)

The address map is **fixed for CSA-101**; students treat it as constant across the course. CSA-201 extends in three directions:

- **M-extension peripherals** (`mul`, `div`, `mod` accelerators) at `0x80160000..0x801FFFFF` (the 6th-15th slave-window slots).
- **Privileged-mode CSRs** (Control & Status Registers per RISC-V Privileged Architecture v1.12): `mstatus`, `mtvec`, `mepc`, `mcause`, etc., these live at CSR-space addresses (`csrr`/`csrw` instructions; not in MMIO space). The MMIO map is unaffected; only the CPU's CSR file grows.
- **PLIC (Platform-Level Interrupt Controller)** at `0x0C000000..0x0FFFFFFF`. Sits at the top of the lower 256 MiB to align with SiFive convention. CSA-201's interrupt-handling track wires the doorbell-pattern IP outputs into the PLIC's interrupt-source ports.

CSA-101 students do not encounter these. The CSA-201 fork introduces them one at a time; the canonical CSA-101 map above is the baseline against which CSA-201 deviations are visible.

---

§5: Cross-references

- **`handouts/cross-chapter-doorbell-scratchpad-descriptor.md`**, address map rationale and 3-pattern interaction overview (companion handout).
- **`handouts/cross-chapter-doorbell-scratchpad-descriptor.md`**. Companion handout; the three IP-Pack interaction patterns each base-address window supports.
- **`handouts/cross-chapter-stdlib-service-reference.md`**, OS-side stdlib contract; per-module MMIO register layouts (the full layouts; this handout summarises them).
- **`handouts/cross-chapter-instr-mem-layout.md`**. Instr_mem + data_mem layout (the BRAM-side companion to this MMIO-side map).
- **Ch 5 §5.7**, Memory-Mapped I/O chapter prose; introduces the routing rule + AXI handshake.
- **Ch 12 §12.6 + §12.8**. Driver-layer + audio-driver chapter prose; uses each base address operationally.

§6: Source-of-truth verification (audit-cycle discipline)

Drift between this handout and the live HDL is a curriculum bug. The audit cycle catches drift via these reconciliation points:

This handout claims	Verified against
HDMI base `0x80100000`	`peripheral-ip-pack/sw/hdmi_demo/hdmi_tile_map.h:5` + `hdl/hdmi/hdmi_axi.v:13`
Audio base `0x80110000`	`peripheral-ip-pack/sw/audio_demo/audio.h:5` + `hdl/audio/audio_pwm_axi.v:13`
PS/2 base `0x80120000`	`peripheral-ip-pack/sw/ps2_demo/ps2_keyboard.h:5` + `hdl/ps2/ps2_keyboard_axi.v:13`
GPIO base `0x80130000`	`peripheral-ip-pack/sw/gpio_demo/gpio.h:5` + `hdl/gpio/gpio_axi.v:13`
DS2 base `0x80140000`	M0.5 deferred. Base reserved.
fpga_pio base `0x80150000`	Pending specification.
Manifold address-decode logic	`peripheral-ip-pack/hdl/manifold/axi_manifold.v`

Discipline: when you notice a discrepancy between this handout and a source-of-truth row above, raise it with your instructor. Handout drift from the live HDL is a curriculum bug; the audit cycle catches and corrects it.
