AXI4-Lite Custom Peripheral Specification Document

Project: AXI4-Lite Custom PL Peripheral for Zynq Blackboard

---

1. Document Overview

1.1 Purpose

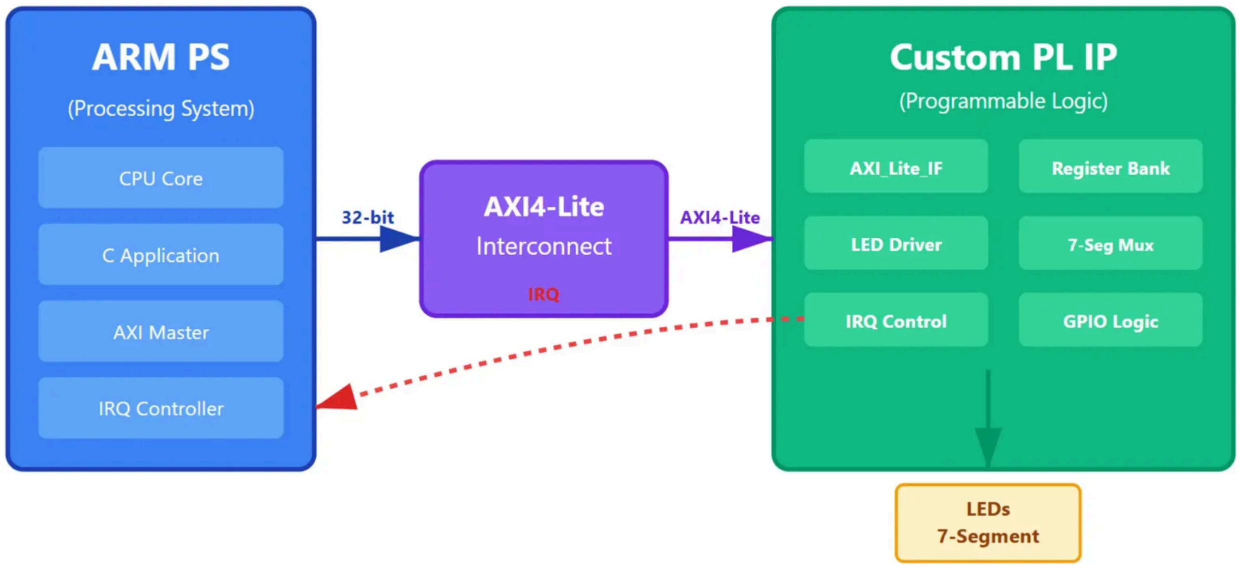
This document provides complete design specifications for the AXI4-Lite custom peripheral IP core intended for integration with the Zynq Processing System (PS). The peripheral exposes memory-mapped registers to control LEDs, seven-segment displays, and manage interrupts from the Programmable Logic (PL) to the ARM processor.

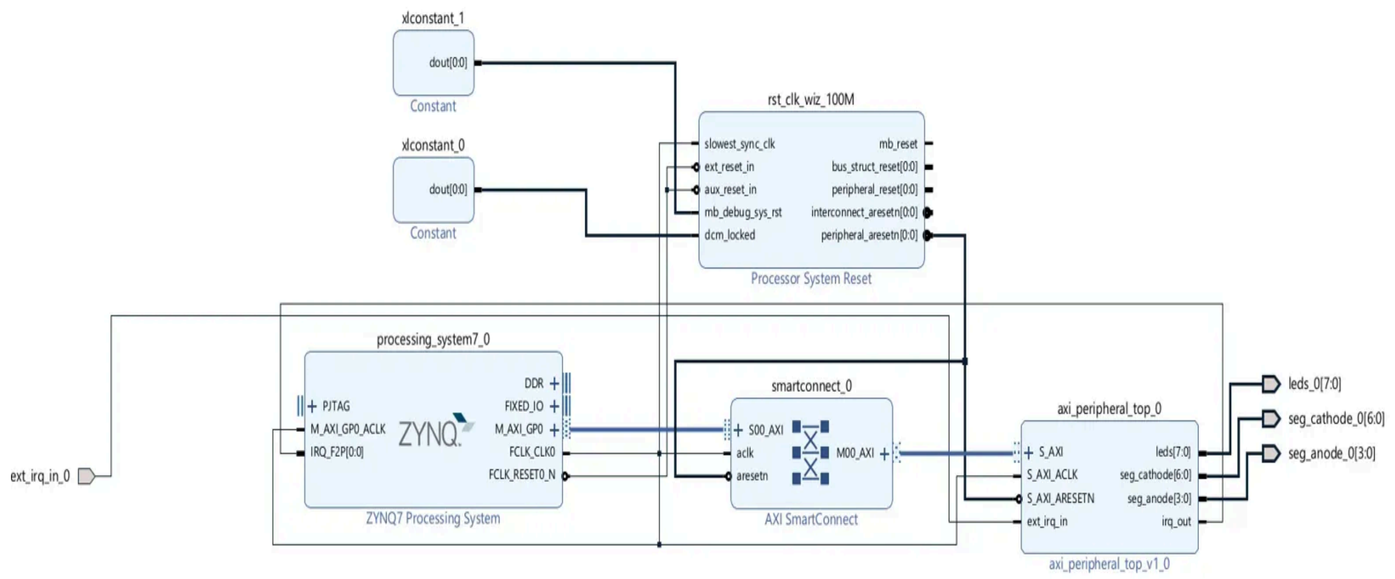
---

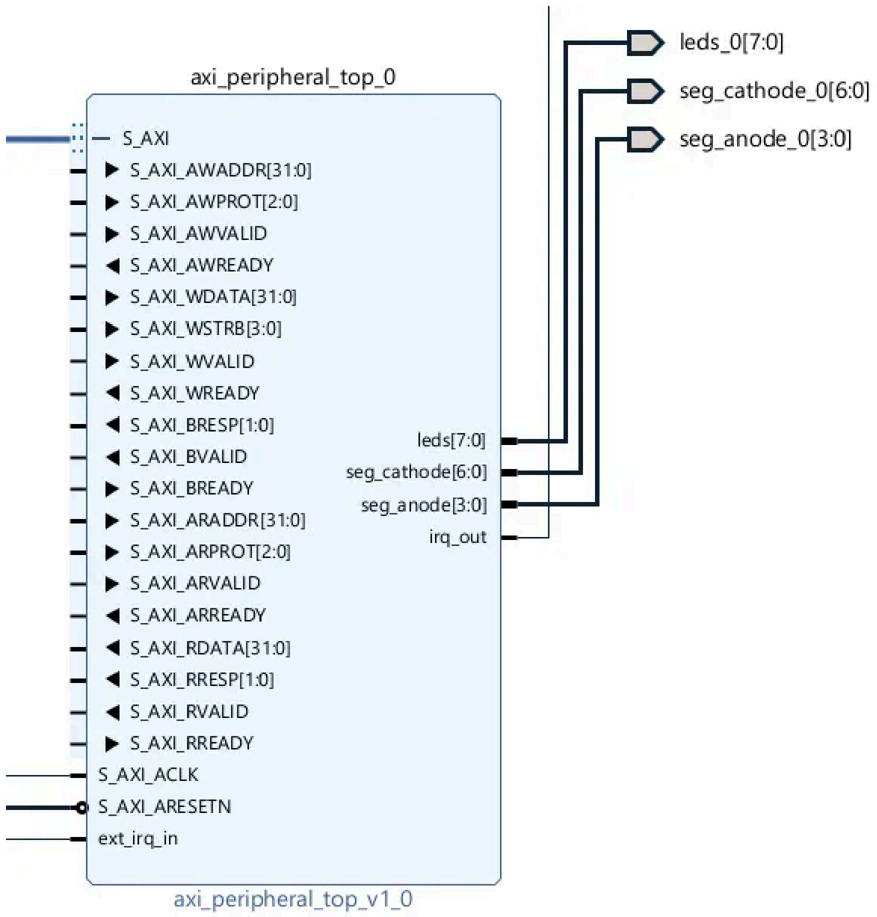
2. System Architecture

2.1 Block Diagram Overview

The peripheral consists of five primary modules integrated under `axi_peripheral_top`:




xlconstant_1
dout[0:0]
Constant
xlconstant_0
dout[0:0]
Constant
rst_clk_wiz_100M
slowest_sync_clk
ext_reset_in
aux_reset_in
mb_debug_sys_rst
dcm_locked
mb_reset
bus_struct_reset[0:0]
peripheral_reset[0:0]
interconnect_aresetn[0:0]
peripheral_aresetn[0:0]
Processor System Reset
processing_system7_0
PJTAG
M_AXI_GP0_ACLK
IRQ_F2P[0:0]
ZYNQ
DDR
FIXED_IO
M_AXI_GP0
FCLK_CLK0
FCLK_RESET0_N
ZYNQ7 Processing System
ext_irq_in_0
smartconnect_0
S00_AXI
aclk
aresetn
M00_AXI
AXI SmartConnect
axi_peripheral_top_0
S_AXI
S_AXI_ACLK
S_AXI_ARESETN
ext_irq_in
leds[7:0]
seg_cathode[6:0]
seg_anode[3:0]
irq_out
axi_peripheral_top_v1_0
leds_0[7:0]
seg_cathode_0[6:0]
seg_anode_0[3:0]


axi_peripheral_top_0
S_AXI
S_AXI_AWADDR[31:0]
S_AXI_AWPROT[2:0]
S_AXI_AWVALID
S_AXI_AWREADY
S_AXI_WDATA[31:0]
S_AXI_WSTRB[3:0]
S_AXI_WVALID
S_AXI_WREADY
S_AXI_BRESP[1:0]
S_AXI_BVALID
S_AXI_BREADY
S_AXI_ARADDR[31:0]
S_AXI_ARPROT[2:0]
S_AXI_ARVALID
S_AXI_ARREADY
S_AXI_RDATA[31:0]
S_AXI_RRESP[1:0]
S_AXI_RVALID
S_AXI_RREADY
S_AXI_ACLK
S_AXI_ARESETN
ext_irq_in
leds[7:0]
seg_cathode[6:0]
seg_anode[3:0]
irq_out
leds_0[7:0]
seg_cathode_0[6:0]
seg_anode_0[3:0]
axi_peripheral_top_v1_0

2.2 Module Hierarchy

1. **axi_peripheral_top** - Top-level wrapper
   - **axi_lite_if** - AXI4-Lite slave interface controller
   - **reg_bank** - Register bank with decode logic
   - **led_driver** - LED control with optional PWM
   - **sevenseg_mux** - 4-digit seven-segment multiplexer
   - **irq_ctrl** - Interrupt controller with debouncing

2.3 Design Parameters

Parameter	Default Value	Description
C_S_AXI_DATA_WIDTH	32	AXI data bus width (bits)
C_S_AXI_ADDR_WIDTH	32	AXI address bus width (bits)
CLK_FREQ_HZ	100,000,000	System clock frequency (Hz)
NUM_LEDS	8	Number of LED outputs
REFRESH_RATE_HZ	1000	Seven-segment refresh rate per digit (Hz)
DEBOUNCE_MS	1	Interrupt debounce time (ms)

---

3. AXI4-Lite Interface Specification

3.1 Functional Description

The `axi_lite_if` module implements a compliant AXI4-Lite slave interface supporting:

- 32-bit aligned register accesses
- Independent read and write channels
- Single outstanding transaction per channel
- OKAY and SLVERR response types
- Full handshaking protocol compliance

3.2 AXI4-Lite Port Description

3.2.1 Global Signals

Port Name	Direction	Width	Description
S_AXI_ACLK	Input	1	AXI clock (synchronous to all signals)
S_AXI_ARESETN	Input	1	Active-low asynchronous reset

3.2.2 Write Address Channel

Port Name	Direction	Width	Description
S_AXI_AWADDR	Input	32	Write address
S_AXI_AWPROT	Input	3	Protection type (unused)
S_AXI_AWVALID	Input	1	Write address valid
S_AXI_AWREADY	Output	1	Write address ready

3.2.3 Write Data Channel

Port Name	Direction	Width	Description
S_AXI_WDATA	Input	32	Write data
S_AXI_WSTRB	Input	4	Write byte strobes
S_AXI_WVALID	Input	1	Write data valid
S_AXI_WREADY	Output	1	Write data ready

3.2.4 Write Response Channel

Port Name	Direction	Width	Description
S_AXI_BRESP	Output	2	Write response (00=OKAY)
S_AXI_BVALID	Output	1	Write response valid
S_AXI_BREADY	Input	1	Write response ready

3.2.5 Read Address Channel

Port Name	Direction	Width	Description
S_AXI_ARADDR	Input	32	Read address
S_AXI_ARPROT	Input	3	Protection type (unused)
S_AXI_ARVALID	Input	1	Read address valid
S_AXI_ARREADY	Output	1	Read address ready

3.2.6 Read Data Channel

Port Name	Direction	Width	Description
S_AXI_RDATA	Output	32	Read data
S_AXI_RRESP	Output	2	Read response (00=OKAY)
S_AXI_RVALID	Output	1	Read data valid
S_AXI_RREADY	Input	1	Read data ready

3.3 Write Channel State Machine

3.3.1 States

- **W_IDLE (2'b00):** Waiting for write address or data
- **W_WAIT_DATA (2'b01):** Address received, waiting for data
- **W_WAIT_ADDR (2'b10):** Data received, waiting for address
- **W_RESPOND (2'b11):** Sending write response

3.3.2 State Transitions

```
W_IDLE:
  - If AWVALID & WVALID → W_RESPOND
  - If AWVALID only → W_WAIT_DATA
  - If WVALID only → W_WAIT_ADDR
  - Else → W_IDLE

W_WAIT_DATA:
  - If WVALID → W_RESPOND
  - Else → W_WAIT_DATA

W_WAIT_ADDR:
  - If AWVALID → W_RESPOND
  - Else → W_WAIT_ADDR
```

W_RESPOND:
- If BREADY → W_IDLE
- Else → W_RESPOND

3.3.3 Write Timing Behavior

- AWREADY and WREADY asserted when respective VALID received
- Both channels can complete in any order
- Register write occurs when both address and data are captured
- BRESP = 2'b00 (OKAY) for all transactions
- BVALID asserted after write completes, held until BREADY

3.4 Read Channel State Machine

3.4.1 States

- **R_IDLE (2'b00):** Waiting for read address
- **R_WAIT_DATA (2'b01):** Address captured, waiting for register data
- **R_RESPOND (2'b10):** Sending read response

3.4.2 State Transitions

R_IDLE:
- If ARVALID → R_WAIT_DATA
- Else → R_IDLE

R_WAIT_DATA:
- If reg_read_valid → R_RESPOND
- Else → R_WAIT_DATA

R_WAIT_DATA:
- If RREADY → R_IDLE
- Else → R_RESPOND

3.4.3 Read Timing Behavior

- ARREADY asserted for one cycle when ARVALID detected
- reg_read_en pulsed to register bank for one cycle
- Read data latched when reg_read_valid asserts
- RVALID held until master asserts RREADY
- RRESP = 2'b00 (OKAY) for all reads
- One cycle latency from address to data availability

3.5 Register Interface Signals

Signal Name	Direction	Width	Description
reg_write_en	Output	1	Register write enable (1 cycle pulse)
reg_write_addr	Output	32	Register write address
reg_write_data	Output	32	Register write data
reg_write_strb	Output	4	Byte write enables
reg_read_en	Output	1	Register read enable (1 cycle pulse)
reg_read_addr	Output	32	Register read address
reg_read_data	Input	32	Register read data
reg_read_valid	Input	1	Register read data valid

4. Register Bank Specification

4.1 Functional Description

The `reg_bank` module implements a memory-mapped register file with the following features:

- Five 32-bit registers with distinct functions
- Byte-level write granularity via write strobes
- Single-cycle read latency
- Edge-triggered interrupt capture
- Automatic interrupt status management

4.2 Memory Map

Address	Register Name	Access	Description
0x00	LED_CTRL	RW	LED control register [7:0]
0x04	SEG_DATA	RW	Seven-segment data [15:0]
0x08	IRQ_ENABLE	RW	Interrupt enable [0]
0x0C	IRQ_STATUS	RO	Interrupt status [0]
0x10	IRQ_CLEAR	WO	Interrupt clear [0]

4.3 Register Descriptions

4.3.1 LED_CTRL (0x00)

Type: Read/Write
Reset Value: 0x00
Width: 8 bits (bits [7:0] of 32-bit word)

Bits	Name	Access	Reset	Description
[7:0]	LED_CONTROL	RW	0x00	LED control bits (1=ON, 0=OFF)
[31:8]	Reserved	RO	0x000000	Reads as 0

Byte Strobe Behavior:

- WSTRB[0]: Controls write to LED_CTRL[7:0]
- WSTRB[3:1]: Ignored

4.3.2 SEG_DATA (0x04)

Type: Read/Write
Reset Value: 0x0000
Width: 16 bits (bits [15:0] of 32-bit word)

Bits	Name	Access	Reset	Description
[3:0]	DIGIT_0	RW	0x0	Hex value for rightmost digit
[7:4]	DIGIT_1	RW	0x0	Hex value for digit 1
[11:8]	DIGIT_2	RW	0x0	Hex value for digit 2
[15:12]	DIGIT_3	RW	0x0	Hex value for leftmost digit
[31:16]	Reserved	RO	0x0000	Reads as 0

Byte Strobe Behavior:

- WSTRB[0]: Controls write to SEG_DATA[7:0]
- WSTRB[1]: Controls write to SEG_DATA[15:8]
- WSTRB[3:2]: Ignored

4.3.3 IRQ_ENABLE (0x08)

Type: Read/Write
Reset Value: 0x0
Width: 1 bit (bit [0] of 32-bit word)

Bits	Name	Access	Reset	Description
[0]	IRQ_EN	RW	0	Interrupt enable (1=enabled, 0=masked)
[31:1]	Reserved	RO	0x00000000	Reads as 0

Byte Strobe Behavior:

- WSTRB[0]: Controls write to IRQ_ENABLE[0]
- WSTRB[3:1]: Ignored

4.3.4 IRQ_STATUS (0x0C)

Type: Read-Only
Reset Value: 0x0
Width: 1 bit (bit [0] of 32-bit word)

Bits	Name	Access	Reset	Description
[0]	IRQ_PENDING	RO	0	Interrupt pending status (1=pending)
[31:1]	Reserved	RO	0x00000000	Reads as 0

Behavior:

- Set to 1 on positive edge of debounced ext_interrupt
- Cleared by write to IRQ_CLEAR
- Read-only from software perspective

4.3.5 IRQ_CLEAR (0x10)

Type: Write-Only
Reset Value: N/A
Width: 1 bit (bit [0] of 32-bit word)

Bits	Name	Access	Reset	Description
[0]	CLEAR	WO	N/A	Write 1 to clear IRQ_STATUS
[31:1]	Reserved	WO	N/A	Ignored

Behavior:

- Writing 1 to bit[0] clears IRQ_STATUS
- Writing 0 has no effect
- Register does not store value (write-only trigger)
- Reads from this address return 0x00000000

Byte Strobe Behavior:

- WSTRB[0]: Must be 1 to clear interrupt
- WSTRB[3:1]: Ignored

4.4 Interrupt Logic

4.4.1 Interrupt Capture

The interrupt status register captures positive edges on the `ext_interrupt` signal:

```
ext_interrupt_d1 ← ext_interrupt @ clk
ext_interrupt_d2 ← ext_interrupt_d1 @ clk
ext_interrupt_posedge = ext_interrupt_d1 & ~ext_interrupt_d2

If ext_interrupt_posedge:
   IRQ_STATUS ← 1
```

4.4.2 Interrupt Output

irq_out = IRQ_STATUS & IRQ_ENABLE

4.4.3 Interrupt Clear

```
If (write_en & write_addr==IRQ_CLEAR & WSTRB[0] & WDATA[0]):
   IRQ_STATUS ← 0
```

Priority: Clear takes precedence over set in the same cycle.

4.5 Register Bank Port Description

Port Name	Direction	Width	Description
clk	Input	1	System clock
resetn	Input	1	Active-low reset

write_en	Input	1	Write enable pulse
write_addr	Input	32	Write address
write_data	Input	32	Write data
write_strb	Input	4	Byte write enables
read_en	Input	1	Read enable pulse
read_addr	Input	32	Read address
read_data	Output	32	Read data
read_valid	Output	1	Read data valid (1 cycle after read_en)
led_control	Output	8	LED control output to led_driver
seg_data	Output	16	Seven-segment data to sevenseg_mux
ext_interrupt	Input	1	External interrupt input (debounced)
irq_out	Output	1	Interrupt request to PS

5. LED Driver Specification

5.1 Functional Description

The `led_driver` module provides direct control of LED outputs with optional PWM dimming capability. The current implementation uses direct mode (ENABLE_PWM=0).

5.2 Operating Modes

5.2.1 Direct Mode (ENABLE_PWM = 0)

- LED outputs directly reflect led_control register
- No PWM modulation
- Single-cycle latency from register update to output

5.2.2 PWM Mode (ENABLE_PWM = 1)

- LEDs modulated at PWM frequency
- Brightness controlled by pwm_duty parameter
- 8-bit resolution (0-255 duty cycle)

5.3 Port Description

Port Name	Direction	Width	Description
clk	Input	1	System clock
resetn	Input	1	Active-low reset
led_control	Input	NUM_LEDS	LED enable bits from register bank
pwm_duty	Input	8	PWM duty cycle (0-255) - unused in direct mode
leds	Output	NUM_LEDS	Physical LED outputs

5.4 Timing Behavior

Direct Mode:

Cycle 0: led_control changes
Cycle 1: leds output updates

Reset Behavior:

- All LEDs forced to 0 when resetn=0
- Output updates on first cycle after reset release

---

6. Seven-Segment Multiplexer Specification

6.1 Functional Description

The `sevenseg_mux` module drives a 4-digit common-anode seven-segment display with time-multiplexed scanning. Each digit is illuminated sequentially at REFRESH_RATE_HZ (1kHz per digit, 4kHz total scan rate).

6.2 Architecture

Cathode Control (active-low)

Hex	Segments (active-low)	Display
0x0	7'b1000000	0

0x1	7'b1111001	1
0x2	7'b0100100	2
0x3	7'b0110000	3
0x4	7'b0011001	4
0x5	7'b0010010	5
0x6	7'b0000010	6
0x7	7'b1111000	7
0x8	7'b0000000	8
0x9	7'b0010000	9
0xA	7'b0001000	A
0xB	7'b0000011	b
0xC	7'b1000110	C
0xD	7'b0100001	d
0xE	7'b0000110	E
0xF	7'b0001110	F

Anode Control (active-low)

Digit Select	Anode Output	Active Digit
2'b00	4'b1110	Digit 0 (rightmost)
2'b01	4'b1101	Digit 1
2'b10	4'b1011	Digit 2
2'b11	4'b0111	Digit 3 (leftmost)

6.3 Port Description

Port Name	Direction	Width	Description
clk	Input	1	System clock
resetn	Input	1	Active-low reset
seg_data	Input	16	Four hex digits [15:12]=D3, [11:8]=D2, [7:4]=D1, [3:0]=D0
seg_cathode	Output	7	Segment outputs {g,f,e,d,c,b,a} (active-low)
seg_anode	Output	4	Digit selects [3:0] (active-low)

6.4 Reset Behavior

- Counter resets to 0
- Digit selector resets to 0 (Digit 0 active)
- All segments and anodes initially off (all 1's)

---

7. Interrupt Controller Specification

7.1 Functional Description

The `irq_ctrl` module conditions external interrupt inputs through:

1. Two-stage synchronizer (CDC safety)
2. Debounce filter (1ms time window)
3. Positive edge detector
4. Single-cycle pulse generator

7.2 Signal Processing Chain

ext_irq_in → [Sync Stage 1] → [Sync Stage 2] → [Debounce] →
[Stable Signal] → [Edge Detect] → [Pulse Gen] → irq_pulse_out

7.3 Synchronizer

Purpose: Safely capture asynchronous input

Implementation:

```
Cycle N:   ext_irq_sync_1 ← ext_irq_in
Cycle N+1: ext_irq_sync_2 ← ext_irq_sync_1
```

Metastability Protection: Two flip-flop stages prevent metastability propagation

7.4 Debounce Filter

Time Constant: DEBOUNCE_MS = 1ms (configurable parameter)

Clock Cycles:

```
DEBOUNCE_COUNT = (CLK_FREQ_HZ / 1000) × DEBOUNCE_MS
               = (100,000,000 / 1000) × 1
               = 100,000 cycles
```

Algorithm:

```
If (ext_irq_sync_2 ≠ ext_irq_stable):
    If (counter == DEBOUNCE_COUNT):
        ext_irq_stable ← ext_irq_sync_2
        counter ← 0
    Else:
        counter ← counter + 1
Else:
    counter ← 0
```

Behavior:

- Input must remain stable for DEBOUNCE_COUNT cycles to change output
- Any transition resets counter
- Filters mechanical switch bounce and electrical noise

7.5 Edge Detector

Implementation:

```
Cycle N:   ext_irq_d1 ← ext_irq_stable
Cycle N+1: ext_irq_posedge = ext_irq_stable & ~ext_irq_d1
```

Detection: Positive edge only (0→1 transition)

7.6 Pulse Generator

Output: Single-cycle pulse on detected edge

Cycle N: Edge detected → irq_pulse_reg ← 1
Cycle N+1: irq_pulse_reg ← 0

7.7 Port Description

Port Name	Direction	Width	Description
clk	Input	1	System clock
resetn	Input	1	Active-low reset
ext_irq_in	Input	1	Raw external interrupt (asynchronous)
irq_pulse_out	Output	1	Clean, debounced pulse (1 cycle)

7.8 Timing Characteristics

Parameter	Value	Description
Synchronization Latency	2 cycles	CDC synchronizer delay
Debounce Time	1ms	Minimum stable time required
Edge-to-Pulse Latency	3 cycles	Sync(2) + Edge(1) after debounce
Pulse Width	1 cycle	Output pulse duration

---

8. Top-Level Integration

8.1 Port Description

8.1.1 AXI4-Lite Interface Ports

(See Section 3.2 for complete AXI signal descriptions)

8.1.2 External I/O Ports

Port Name	Direction	Width	Description
leds	Output	8	Physical LED outputs
seg_cathode	Output	7	Seven-segment cathodes {g,f,e,d,c,b,a}
seg_anode	Output	4	Seven-segment anodes (digit selects)
irq_out	Output	1	Interrupt to PS IRQ controller
ext_irq_in	Input	1	External interrupt source (e.g., button)

8.2 Internal Signal Connectivity

axi_lite_if → reg_bank:
- reg_write_en, reg_write_addr, reg_write_data, reg_write_strb
- reg_read_en, reg_read_addr
- reg_read_data, reg_read_valid (from reg_bank)

reg_bank → led_driver:
- led_control[7:0]

reg_bank → sevenseg_mux:
- seg_data[15:0]

irq_ctrl → reg_bank:
- ext_interrupt (irq_pulse_out from irq_ctrl)

reg_bank → top-level:
- irq_out (to PS)

8.3 Clock and Reset Strategy

Single Clock Domain:

- All modules operate on S_AXI_ACLK
- No clock domain crossings within peripheral
- Typical frequency: 100 MHz (from PS FCLK_CLK0)

Reset:

- S_AXI_ARESETN is active-low asynchronous
- Connected to all module reset inputs
- Sourced from PS or external reset logic

9. Functional Requirements

9.1 AXI4-Lite Protocol Compliance

REQ-AXI-001: The peripheral shall implement a fully compliant AXI4-Lite slave interface according to ARM IHI0022E specification.

REQ-AXI-002: The peripheral shall support only 32-bit aligned accesses (address bits [1:0] ignored).

REQ-AXI-003: All AXI transactions shall complete with OKAY response (2'b00) regardless of address validity.

REQ-AXI-004: The peripheral shall support independent write address, write data, and read address channels.

REQ-AXI-005: Write address and write data may arrive in any order and shall be buffered until both are available.

REQ-AXI-006: The peripheral shall support a single outstanding transaction per channel (no pipelining).

REQ-AXI-007: The peripheral shall not assert AWREADY or ARREADY until it can accept the transaction.

REQ-AXI-008: Once BVALID or RVALID is asserted, it shall remain high until corresponding BREADY or RREADY is received.

9.2 Register Bank Requirements

REQ-REG-001: LED_CTRL register shall directly control 8 LED outputs with single-cycle update latency.

REQ-REG-002: SEG_DATA register shall store 4 hexadecimal digits for seven-segment display.

REQ-REG-003: IRQ_ENABLE shall gate the interrupt output when cleared (0).

REQ-REG-004: IRQ_STATUS shall be set on positive edge of external interrupt and cleared only by software write to IRQ_CLEAR.

REQ-REG-005: Writing 1 to IRQ_CLEAR[0] shall clear IRQ_STATUS in the same cycle.

REQ-REG-006: If external interrupt edge and IRQ_CLEAR write occur simultaneously, clear shall take priority.

REQ-REG-007: All reserved bits shall read as 0 and writes shall be ignored.

REQ-REG-008: Byte write strobes shall enable independent byte-level updates within registers.

REQ-REG-009: Read operations shall not have side effects on register contents.

REQ-REG-010: Register read data shall be valid one cycle after read_en assertion.

9.3 LED Driver Requirements

REQ-LED-001: LED outputs shall reflect led_control register state in direct mode.

REQ-LED-002: LED output changes shall occur within 2 clock cycles of register write.

REQ-LED-003: All LEDs shall be off (0) during reset.

REQ-LED-004: PWM mode (if enabled) shall modulate LED brightness based on pwm_duty value.

REQ-LED-005: PWM frequency (if enabled) shall be sufficient to avoid visible flicker (>100 Hz).

9.4 Seven-Segment Display Requirements

REQ-SEG-001: The display shall multiplex 4 digits with refresh rate of 1 kHz per digit (4 kHz total).

REQ-SEG-002: Each digit shall display hexadecimal values 0-F correctly with standard 7-segment encoding.

REQ-SEG-003: Segment outputs shall be active-low for common anode displays.

REQ-SEG-004: Anode outputs shall be active-low with only one digit enabled at any time.

REQ-SEG-005: Display update to new seg_data value shall be visible within one complete scan cycle (4 ms).

REQ-SEG-006: All segments and anodes shall be off during reset.

REQ-SEG-007: Digit scanning shall be continuous and uninterrupted during normal operation.

9.5 Interrupt Controller Requirements

REQ-IRQ-001: External interrupt input shall pass through a 2-stage synchronizer to prevent metastability.

REQ-IRQ-002: Synchronized interrupt shall be debounced with 1 ms time constant (configurable).

REQ-IRQ-003: Only positive edges (0→1 transitions) of the debounced signal shall generate interrupt pulses.

REQ-IRQ-004: Interrupt output pulse shall be exactly one clock cycle wide.

REQ-IRQ-005: Debounce counter shall reset if input changes state before timeout expires.

REQ-IRQ-006: Minimum input stable time of DEBOUNCE_MS required for edge detection.

REQ-IRQ-007: Total latency from external edge to irq_pulse_out shall not exceed DEBOUNCE_MS + 10 clock cycles.

9.6 System Integration Requirements

REQ-SYS-001: All modules shall operate in a single clock domain (S_AXI_ACLK).

REQ-SYS-002: The peripheral shall meet timing at 100 MHz clock frequency.

REQ-SYS-003: Reset shall be asynchronous assertion, synchronous de-assertion.

REQ-SYS-004: All outputs shall be registered to minimize combinatorial paths.

REQ-SYS-005: No combinatorial loops shall exist in the design.

REQ-SYS-006: The design shall synthesize without inferred latches except where explicitly intended.

REQ-SYS-007: All case statements shall have default clauses.

---

10. Known Limitations and Assumptions

10.1 Limitations

1. **Single Outstanding Transaction:** The AXI interface does not support pipelined transactions. Only one read and one write can be outstanding at any time.

2. **No Burst Support:** Only single-beat transactions supported (inherent to AXI4-Lite).

3. **Fixed 32-bit Width:** Data width is fixed at 32 bits; not parameterizable for 64-bit.

4. **No Error Response:** All transactions return OKAY response regardless of address validity. Invalid addresses read as 0x00000000.

5. **LED PWM Disabled:** Current configuration uses direct LED control; PWM feature exists but is disabled.

6. **Debounce Time:** 1ms debounce is optimized for simulation speed; production hardware may require 20-50ms.

7. **No Partial Word Reads:** Reads always return full 32-bit word; byte/halfword reads not distinguished.

10.2 Assumptions

1. **Clock Frequency:** System clock assumed to be 100 MHz from PS FCLK_CLK0.

2. **Reset Source:** ARESETN driven by PS or reliable reset controller.

3. **External Interrupt:** ext_irq_in assumed to be from push-button or similar mechanical switch.

4. **Display Type:** Seven-segment display is common-anode type with active-low segments.

5. **LED Current:** LEDs have appropriate current-limiting resistors externally.

6. **Single Clock Domain:** No clock domain crossing logic needed; all PS-PL communication at same frequency.

7. **Static Configuration:** Parameters set at synthesis time; no runtime reconfiguration.