

S.A.B.E.R. – wifi_pack_reg Product Guide

VLNV: xilinx.com:user:wifi_pack_reg:1.0

Date: August 21, 2025 License: MIT

1. Overview

wifi_pack_reg is a lightweight AXI4-Lite peripheral that exposes a **1792-bit** encrypted frame (56×32 -bit words) to a processor. A single-cycle strobe **send_data_i** latches the full frame from **data_i**[1791:0] into an internal buffer. A status/clear register allows software to check frame readiness and acknowledge/clear the capture.

Typical use Connect **data_i** to the cipher/packer output of the *encryptor_system*, assert **send_data_i** for one cycle when a frame is ready, then read the 56 words via AXI. Finally, write 1 to bit 0 of the status register to produce **clr_o** (clear pulse).

2. Features

- AXI4-Lite slave, 32-bit data, 8-bit address (configurable).
- Captures a full 1792-bit frame atomically on the rising edge of **send_data_i**.
- 56 read-only data words + 1 status/clear word at offset 0xE0.
- MSB-first word ordering: word 0 = **data_i**[1791:1760], word 55 = **data_i**[31:0].
- Write-1-to-clear (W1C) generates a one-cycle **clr_o** pulse.
- **data_valid_i** reflected in the status register (bit 1).
- **Clock domain safety:** **send_data_i** is synchronized to the AXI clock; the bus **data_i** must be stable for at least one AXI cycle around the strobe.

3. Parameters

| Name | Description | Default |
|---------------------|---|---------|
| C_data_o_DATA_WIDTH | AXI data width (must be 32). | 32 |
| C_data_o_ADDR_WIDTH | AXI address width in bits (≥ 8). | 8 |

4. Ports

| Name | Dir | Width | Clock | Description |
|---------------------|-----|-------|-------|--|
| data_i | in | 1792 | AXI | Flattened frame (56×32 b). |
| data_valid_i | in | 1 | AXI | Frame-ready status from packer (mirrored in status bit 1). |
| send_data_i | in | 1 | AXI | Rising edge latches data_i into internal buffer. |
| clr_o | out | 1 | AXI | One-cycle pulse when SW writes 1 to status bit 0. |

AXI4-Lite “data_o” interface (standard signals not expanded here).

5. Memory Map (AXI4-Lite)

Address decoding uses word indices $\text{idx} = \text{AR}/\text{AWADDR}[7:2]$ (for 8-bit address). Total space is **57 words** = 0xE4 bytes.

5.1 Data window (read-only)

| Offset | Idx | Access | Content (MSB-first) |
|--------------|-------|--------|---|
| 0x00 .. 0xDC | 0..55 | R | Word i = <code>data_i[TOTAL_BITS-1 - 32*i -: 32]</code> . |

5.2 Status / Clear register

| Offset | Idx | Access | Description |
|--------|-----|--------|---|
| 0xE0 | 56 | R/W | [31:3] Reserved (read as 0); [2] <code>captured</code> (1 after capture); [1] <code>data_valid_i</code> ; [0] <code>CLR_W1C</code> : write 1 to generate <code>clr_o</code> . Reads as 0. |

Clear behavior Writing a 1 to bit 0 asserts `clr_o` for one AXI clock, and clears `captured`. Writes to other bits are ignored.

6. Word Ordering

The 1792-bit frame is presented **MSB-first** to software:

$$\text{Word0} = \text{data_i}[1791:1760], \dots, \text{Word55} = \text{data_i}[31:0].$$

This matches the packer layout used by the `encryptor_system`. If your software expects a different ordering, reverse the sequence when reading.

7. Timing and CDC Notes

- **Capture:** The rising edge of `send_data_i` (synchronized internally) snapshots all 1792 bits from `data_i` in one AXI clock.
- **Data stability:** Ensure `data_i` is stable for at least one AXI cycle around the strobe. Using the same clock domain is recommended.
- `data_valid_i` is sampled on the AXI clock and reflected in status bit 1.

8. Software Example (C pseudo-code)

```
#define WIFI_BASE 0x40000000u
#define REG(i) (*(volatile uint32_t *) (WIFI_BASE + ((i) << 2)))
#define REG_STATUS REG(56) // 0xE0
#define CLR_W1C (1u << 0)
#define ST_DATA_RDY (1u << 1)
#define ST_CAPTURED (1u << 2)

void read_frame(uint32_t *buf56) {
    // Poll until hardware indicates a fresh capture is present
    while ((REG_STATUS & ST_CAPTURED) == 0) { /* spin or sleep */ }

    // Read 56 words (MSB-first)
    for (int i = 0; i < 56; ++i)
        buf56[i] = REG(i);

    // Optional: check producer status
    uint32_t st = REG_STATUS;
```

```

(void)st; // bit1 mirrors data_valid_i

// Acknowledge/clear
REG_STATUS = CLR_W1C;
}

```

9. Integration Checklist

- Connect AXI4-Lite interface *data_o* to the SoC interconnect at a 4-byte aligned base address.
- Drive `send_data_i` with a *single-cycle* pulse when the 1792-bit frame on `data_i` is valid.
- Tie `data_valid_i` to your packer “frame ready” flag (optional but recommended for software polling).
- Ensure timing closure for the 1792-bit bus into this IP at the AXI clock frequency.

10. Revision History

| Version | Date | Notes |
|---------|-----------------|--|
| 1.0 | August 21, 2025 | Initial release of <code>wifi_pack_reg</code> (AXI4-Lite, 56 data words + status/clear). |

Contact For questions or issues, please open a ticket in your project repository.