

# Zen PPU Debug — Datasheet

## UART — Datasheet (DSHT-00-INDEX)

The **product DATASHEET** for the **Zen UART** (zen-uart-kit-v2) — a 16550-compatible universal asynchronous receiver/transmitter with independent TX/RX FIFOs, on a struct-typed peripheral bus with APB / AXI4-Lite frontends. This is the **row-1 hub** of a hierarchical, multi-file datasheet: how to read it, the chapter map, conventions, and provenance. Product/integrator-facing. Instantiated from shared-agent-infra/DATASHEET-STANDARD.md.  
Product altitude — *what it does + how to integrate*, not *how it's built*. = outline · = authored (pending review) · = content complete + reviewed.

### What this datasheet is (and is not)

- **Is:** the product datasheet — headline features, key specs, standards compliance, operating modes, interface/pinout, programming model, performance, physical characteristics, integration/deliverables.
- **Is not:** the internal design spec (DS-NN),  $\mu$ Arch (UARCH), or DV testplan (DV-NN). This datasheet **distills** those; each chapter cites its source. Ground truth = the as-built RTL (rtl/uart/); **RTL wins** on structural disagreement.
- **Altitude:** *what the UART does + how to use it*. Uncharacterized numbers are marked **TBC** with an owner.

### How to read it

- Start at `[[docs/datasheets/uart/DSHT-01-OVERVIEW|DSHT-01]]`, then read the chapters relevant to your integration.
- **Numbers policy:** every quantitative spec is (a) sourced from the RTL/register-map, (b) a standards requirement, or (c) marked **TBC** with an owner. **No unsourced numbers.**

### Chapter map

| DSHT                                                               | Chapter                             | Scope                                     | (DS)                 |
|--------------------------------------------------------------------|-------------------------------------|-------------------------------------------|----------------------|
| [[docs/datasheets/uart/DSHT-00-INDEX DSHT-00]]                     | Intro & TOC                         | this hub — map · conventions              | —                    |
| [[docs/datasheets/uart/DSHT-01-OVERVIEW DSHT-01]]                  | Product overview                    | positioning · target apps · block diagram | rtl/ua<br>[[doc LICE |
| [[docs/datasheets/uart/DSHT-02-FEATURES-COMPLIANCE DSHT-02]]       | Key features & standards compliance | feature matrix · 16550 compliance         | rtl/ua               |
| [[docs/datasheets/uart/DSHT-03-APPLICATIONS DSHT-03]]              | Applications / use models           | target deployments                        | —                    |
| [[docs/datasheets/uart/DSHT-04-CONFIGURABLE-OPTIONS DSHT-04]]      | Configurable options                | build-time parameters                     | rtl/ua               |
| [[docs/datasheets/uart/DSHT-05-FUNCTIONAL-OVERVIEW DSHT-05]]       | Functional overview                 | black-box behavior · TX/RX/IRQ            | rtl/ua               |
| [[docs/datasheets/uart/DSHT-06-INTERFACES-SIGNALS DSHT-06]]        | Interfaces & signals                | top-level interfaces by group             | rtl/ua               |
| [[docs/datasheets/uart/DSHT-07-CLOCKING-RESET DSHT-07]]            | Clocking & reset                    | clock/reset requirements                  | rtl/ua               |
| [[docs/datasheets/uart/DSHT-08-POWER-ON-RESET-INIT DSHT-08]]       | Power-on / reset / init             | bring-up ordering                         | rtl/ua               |
| [[docs/datasheets/uart/DSHT-09-PROGRAMMING-MODEL DSHT-09]]         | Programming model                   | access model · sequences                  | rtl/ua               |
| [[docs/datasheets/uart/DSHT-10-REGISTER-MAP DSHT-10]]              | Register map                        | CSR offsets/fields/reset/access           | zen_ · rtl/u         |
| [[docs/datasheets/uart/DSHT-11-PERFORMANCE DSHT-11]]               | Performance                         | throughput/latency envelope               | rtl/ua               |
| [[docs/datasheets/uart/DSHT-12-PHYSICAL DSHT-12]]                  | Physical / implementation           | area/fmax · constraints                   | TBC                  |
| [[docs/datasheets/uart/DSHT-13-RAS DSHT-13]]                       | RAS & data integrity                | error reporting summary                   | rtl/ua               |
| [[docs/datasheets/uart/DSHT-14-VERIFICATION-DELIVERABLES DSHT-14]] | Verification & deliverables         | TB/coverage · manifest                    | verif<br>[[doc LICE  |
| [[docs/datasheets/uart/DSHT-15-INTEGRATION-GUIDE DSHT-15]]         | Integration guide                   | instantiation · tie-offs · DFT            | rtl/ua               |
| [[docs/datasheets/uart/DSHT-16-ORDERING-LICENSING DSHT-16]]        | Ordering / licensing / support      | commercial                                | [[doc LICE           |
| [[docs/datasheets/uart/DSHT-A-APPENDICES DSHT-A]]                  | Appendices                          | signals · acronyms · revision history     | —                    |

## Moniker & provenance

- **Moniker DSHT** (DataSheeT) per shared-agent-infra/DATASHEET-STANDARD.md.
- **Path:** docs/datasheets/uart/DSHT-NN-NAME.md; hub = DSHT-00-INDEX.md; appendices = DSHT-A-APPENDICES.md.
- **Provenance:** distilled from the as-built RTL (rtl/uart/zen\_uart.sv, zen\_uart\_apb.sv, zen\_uart\_axil.sv) + register enum (rtl/include/zen\_periph\_pkg.sv) + the UART license kit. Product numbers reconcile against the RTL; **RTL wins** on structural disagreement.

**Scope.** What the Zen UART is, who it's for, and its headline characteristics. Product altitude. Distills the as-built RTL (rtl/uart/) + [[docs/kits/UART-LICENSE-KIT|UART license kit]].

- The **Zen UART** is a 16550-compatible asynchronous serial controller: full-duplex TX + RX with independent FIFOs, programmable frame format and baud rate, and a single aggregated interrupt.
- It attaches to the common Zen peripheral bus (`periph_req_t` / `periph_rsp_t`) and is delivered with **APB4** and **AXI4-Lite** frontends (`zen_uart_apb`, `zen_uart_axil`) for drop-in SoC integration.
- Software-compatible with the ubiquitous 16550 register model, so existing UART drivers and boot ROMs work unchanged.

| Characteristic     | Value                              | Source                                       |
|--------------------|------------------------------------|----------------------------------------------|
| Register model     | 16550-compatible, 8 registers      | zen_periph_pkg.sv (SOURCED)                  |
| TX / RX FIFO depth | FIFO_DEPTH = 16 (default)          | rtl/uart/zen_uart.sv:18 (SOURCED)            |
| Data bits          | 5 / 6 / 7 / 8                      | LCR (SOURCED)                                |
| Parity             | none / odd / even (stick optional) | LCR (SOURCED)                                |
| Stop bits          | 1 / 2                              | LCR (SOURCED)                                |
| Reference clock    | CLK_FREQ = 100 MHz (default, doc)  | rtl/uart/zen_uart.sv:20 (SOURCED)            |
| Max baud           | function of CLK_FREQ / divisor     | TBC (owner: verif)                           |
| Bus frontends      | native periph · APB4 · AXI4-Lite   | rtl/uart/ (SOURCED)                          |
| Interrupt          | 1 aggregated line (irq_o)          | rtl/uart/zen_uart.sv:41 (SOURCED)            |
| Verification       | 10/10 assertion-based tests        | [[docs/kits/UART-LICENSE-KIT kit]] (SOURCED) |

The diagram illustrates the Zen UART peripheral's internal structure and signal flow. At the top, the peripheral is identified as 'periph\_req/rsp (or APB4 / AXI4-Lite frontend)'. The main data path consists of a 'Zen UART' block, which is divided into a TX path and an RX path. The TX path includes a 'TX FIFO' and a 'TX shift' register, with data flowing from the FIFO to the shift register and then to the 'uart\_tx\_o' output. The RX path includes an 'RX FIFO' and an 'RX shift' register, with data flowing from the 'uart\_rx\_i' input to the shift register and then to the FIFO. Control and status signals are shown at the bottom: 'THR' (Transmit Holding Register) is an input to the TX FIFO; 'RBR' (Receive Buffer Register) is an output from the RX FIFO; 'baud-gen · LCR/IER/IIR/FCR · IRQ agg' is a combined control/status signal; and 'irq\_o' is the interrupt output signal.

## Standards compliance

- **NS16550 / 16550A** register and behavioral model (de-facto industry standard). See [\[\[docs/datasheets/uart/DSHT-02-FEATURES-COMPLIANCE|DSHT-02\]\]](#).

## Source cross-refs

- RTL: [rtl/uart/zen\\_uart.sv](#), [zen\\_uart\\_apb.sv](#), [zen\\_uart\\_axil.sv](#) · Registers: [rtl/include/zen\\_periph\\_pkg.sv](#) · Kit: [\[\[docs/kits/UART-LICENSE-KIT|UART-LICENSE-KIT\]\]](#) · Catalog: [\[\[docs/catalog/IP-CATALOG-04-SERIAL-IO|vol. 04\]\]](#).

## DSHT-01 · UART · product overview.

title: “UART — Key Features & Standards Compliance” created: 2026-09-14 status: authored (pending review) standard: shared-agent-infra/DATASHEET-STANDARD.md —

# DSHT-02 · Key Features & Standards Compliance

**Scope.** The feature matrix and the 16550 compliance checklist. Distills [rtl/uart/zen\\_uart.sv](#).

## Feature matrix

| Feature                            | Supported | Notes                         | Source  |
|------------------------------------|-----------|-------------------------------|---------|
| Full-duplex TX + RX                |           | independent datapaths         | RTL     |
| TX / RX FIFOs                      |           | depth FIFO_DEPTH (16 default) | RTL:18  |
| Programmable data bits (5–8)       |           | via LCR                       | RTL     |
| Parity none/odd/even               |           | via LCR                       | RTL     |
| 1 / 2 stop bits                    |           | via LCR                       | RTL     |
| Programmable baud (DLAB divisor)   |           | DLL/DLM via DLAB              | RTL     |
| Line-status reporting (LSR)        |           | OE/PE/FE/BI, THRE, DR         | RTL     |
| Modem control/status (MCR/MSR)     |           | register-modeled              | RTL     |
| Scratch register (SCR)             |           |                               | RTL     |
| Aggregated interrupt (IIR-decoded) |           | single irq_o                  | RTL:41  |
| RX metastability synchronizer      |           | 2-FF sync on uart_rx_i        | RTL:266 |
| Hardware auto-flow (RTS/CTS)       |           | not implemented (TBC roadmap) | RTL     |
| DMA request lines                  |           | FIFO-level IRQ only           | RTL     |

## Standards compliance

| Standard                              | Scope                                                         | Status     |
|---------------------------------------|---------------------------------------------------------------|------------|
| NS16550 / 16550A register model       | RBR/THR, IER, IIR/FCR, LCR, MCR, LSR, MSR, SCR + DLAB divisor | compatible |
| Frame format (start/data/parity/stop) | async serial framing                                          |            |

## Notes

- Auto-flow-control (RTS/CTS handshake) and DMA handshake are candidate extensions; today flow is managed by software via LSR/MSR polling or interrupt.

## Source cross-refs

- RTL: rtl/uart/zen\_uart.sv · Registers: rtl/include/zen\_periph\_pkg.sv.

## DSHT-02 · UART · features & compliance.

title: “UART — Applications / Use Models” created: 2026-09-14 status: authored  
(pending review) standard: shared-agent-infra/DATASHEET-STANDARD.md —

## DSHT-03 · Applications / Use Models

| **Scope.** Target deployments and typical use models for the Zen UART.

### Target applications

- **Console / debug UART** — the boot and bring-up serial console on any Zen SoC (zen-cpu, zen-noc).
- **Host / MCU comms** — general-purpose async link to companion controllers or external hosts.
- **Legacy peripheral bridge** — GPS, modems, and legacy RS-232-class devices via an off-chip transceiver.

### Use models

- **Polled** — driver reads LSR (DR, THRE) and moves bytes through RBR/THR. Simplest bring-up path.
- **Interrupt-driven** — enable sources in IER; service on irq\_o, decode cause from IIR. FIFO trigger level set via FCR.
- **Boot ROM** — 16550 compatibility lets standard boot ROM / firmware UART code run unmodified.

### Recommended companions

- [[docs/catalog/IP-CATALOG-02-INTERRUPTS|Interrupt controller (PLIC/AIA)] for irq\_o routing · [[docs/catalog/IP-CATALOG-01-BUS-FABRIC|Bus Fabric]] for APB/AXI4-Lite attach.

## Source cross-refs

- Kit: [[docs/kits/UART-LICENSE-KIT|UART-LICENSE-KIT]].

## DSHT-03 · UART · applications.

title: “UART — Configurable Options” created: 2026-09-14 status: authored  
(pending review) standard: shared-agent-infra/DATASHEET-STANDARD.md —

## DSHT-04 · Configurable Options

| **Scope.** The build-time (parameter) and run-time (register) options exposed to the integrator. Distills rtl/uart/zen\_uart.sv.

### Build-time parameters

Source: RTL module header zen\_uart — SOURCED.

| Parameter  | Type | Default     | Range                            | Description                                                                                                  |
|------------|------|-------------|----------------------------------|--------------------------------------------------------------------------------------------------------------|
| FIFO_DEPTH | int  | 16          | power-of-2, $\geq 2$ (TBC upper) | Depth of both TX and RX FIFOs: $\text{FIFO\_PTR\_WIDTH} = \lceil \log_2(\text{FIFO\_DEPTH}) \rceil$ derived. |
| CLK_FREQ   | int  | 100_000_000 | design-dependent                 | Reference clock in Hz. Documentation/aid value; baud divisor is programmed at run time                       |

### Run-time options (register-selected)

- **Frame format** — data bits (5–8), parity (none/odd/even), stop bits (1/2): **LCR**.
- **Baud rate** — divisor latch (DLL/DLM) accessed via **DLAB** bit in LCR.
- **Interrupt sources** — **IER** (RX-data-available, THR-empty, line-status, modem-status).
- **FIFO control / trigger** — **FCR**.
- Full field detail: [[docs/datasheets/uart/DSHT-10-REGISTER-MAPIDSHT-10]].

## Source cross-refs

- RTL: rtl/uart/zen\_uart.sv:18,20,47 · Registers: rtl/include/zen\_periph\_pkg.sv.

## *DSHT-04 · UART · configurable options.*

title: “UART — Functional Overview” created: 2026-09-14 status: authored  
(pending review) standard: shared-agent-infra/DATASHEET-STANDARD.md —

## DSHT-05 · Functional Overview

**Scope.** Black-box behavior of the Zen UART — TX path, RX path, baud generation, and interrupt aggregation. Product altitude; internals summarized, not duplicated. Distills rtl/uart/zen\_uart.sv.

### Transmit path

- Software writes bytes to **THR** (RBR/THR offset, DLAB=0); bytes enqueue into the TX FIFO (depth FIFO\_DEPTH).
- The TX shift engine (FSM) serializes each byte per the LCR frame format: start bit → data bits (LSB first) → optional parity → stop bit(s).
- **LSR.THRE** indicates the holding register/FIFO can accept more data.

### Receive path

- `uart_rx_i` is passed through a 2-FF metastability synchronizer (rtl/uart/zen\_uart.sv:266) before sampling.
- The RX shift engine samples the frame, checks parity/framing, and enqueues the byte into the RX FIFO.
- **LSR.DR** indicates received data is available at **RBR**; **LSR** also reports overrun (OE), parity (PE), framing (FE), and break (BI).

### Baud generation

- A divisor latch (DLL/DLM), accessed via the LCR **DLAB** bit, divides the reference clock to the bit-rate tick. Achievable baud =  $\text{CLK\_FREQ} / (16 \times \text{divisor})$  per the 16550 model. Exact max baud is **TBC** (owner: verif).

### Interrupt aggregation

- Enabled sources (IER) are prioritized and encoded in **IIR**; the aggregate drives the single `irq_o` line. Reading IIR/updating FCR manages the cause and FIFO trigger.

## Source cross-refs

- RTL FSMs: rtl/uart/zen\_uart.sv (tx\_state @149, rx\_state @284) · Registers: rtl/include/zen\_periph\_pkg.sv.

## *DSHT-05 · UART · functional overview.*

title: “UART — Interfaces & Signals” created: 2026-09-14 status: authored  
(pending review) standard: shared-agent-infra/DATASHEET-STANDARD.md note:  
“authored from datasheet-gen output DSHT-06-INTERFACES-SIGNALS.gen.md  
(RTL-sourced)” —

## DSHT-06 · Interfaces & Signals

**Scope.** Top-level interfaces of `zen_uart` by group. Every port RTL-sourced from rtl/uart/zen\_uart.sv; full port table in [[docs/datasheets/uart/DSHT-A-APPENDICESIAApp A]].

## Interface groups

| Group          | Signals                                    | Description                                      |
|----------------|--------------------------------------------|--------------------------------------------------|
| Clock / reset  | clk_i, rst_ni                              | single clock; active-low async reset             |
| Peripheral bus | req_i : periph_req_t, rsp_o : periph_rsp_t | struct-typed req/rsp; byte access (wstrb unused) |
| Serial pins    | uart_tx_o, uart_rx_i                       | async serial TX/RX (to off-chip transceiver)     |
| Interrupt      | irq_o                                      | single aggregated interrupt                      |

## Port table (RTL-sourced)

Source: rtl/uart/zen\_uart.sv module header — SOURCED.

| Direction | Signal    | Type / width | Notes                                      |
|-----------|-----------|--------------|--------------------------------------------|
| input     | clk_i     | logic        | reference clock                            |
| input     | rst_ni    | logic        | active-low reset                           |
| input     | req_i     | periph_req_t | bus request; wstrb bits unused (byte-only) |
| output    | rsp_o     | periph_rsp_t | bus response                               |
| output    | uart_tx_o | logic        | serial transmit                            |
| input     | uart_rx_i | logic        | serial receive (synchronized internally)   |
| output    | irq_o     | logic        | aggregated interrupt                       |

## Bus frontends

- **APB4:** zen\_uart\_apb — APB4 slave wrapper.
- **AXI4-Lite:** zen\_uart\_axil — AXI4-Lite slave wrapper.
- Both adapt the host bus to the native periph\_req\_t/periph\_rsp\_t contract.

## Standards compliance

- Bus frontends conform to ARM **APB4** / **AXI4-Lite**; serial framing per 16550 (see [[docs/datasheets/uart/DSHT-02-FEATURES-COMPLIANCEIDSH-02]]).

## Source cross-refs

- RTL: rtl/uart/zen\_uart.sv, zen\_uart\_apb.sv, zen\_uart\_axil.sv · gen: DSHT-06-INTERFACES-SIGNALS.gen.md.

## DSHT-06 · UART · interfaces & signals.

title: “UART — Clocking & Reset” created: 2026-09-14 status: authored (pending review) standard: shared-agent-infra/DATASHEET-STANDARD.md —

## DSHT-07 · Clocking & Reset

| **Scope.** External clock and reset requirements. Distills rtl/uart/zen\_uart.sv.

### Clocking

| Item                | Spec                       | Source           |
|---------------------|----------------------------|------------------|
| Clock inputs        | 1 (clk_i)                  | RTL:23 (SOURCED) |
| Clock domains       | single synchronous domain  | RTL (SOURCED)    |
| Reference frequency | CLK_FREQ (100 MHz default) | RTL:20 (SOURCED) |
| Baud derivation     | clk_i ÷ (16 × divisor)     | 16550 model      |
| Max fmax            | TBC (owner: PD)            | —                |

## Reset

| Item             | Spec                                                                                                             | Source           |
|------------------|------------------------------------------------------------------------------------------------------------------|------------------|
| Reset input      | rst_ni, active-low                                                                                               | RTL:24 (SOURCED) |
| Reset style      | asynchronous assert,<br>synchronous release<br>recommended                                                       | TBC (owner: PD)  |
| Post-reset state | FIFOs empty, registers at<br>reset values (see<br>[[docs/datasheets/uart/DSHT-<br>10-REGISTER-MAPIDSHT-<br>10]]) | RTL              |

## CDC notes

- uart\_rx\_i is asynchronous to clk\_i and is synchronized internally (2-FF, RTL:266). No other external CDC required for single-clock integration.

## Source cross-refs

- RTL: rtl/uart/zen\_uart.sv:20,23,24,266.

## DSHT-07 · UART · clocking & reset.

title: “UART — Power-On / Reset / Init Sequence” created: 2026-09-14 status: authored (pending review) standard: shared-agent-infra/DATASHEET-STANDARD.md —

## DSHT-08 · Power-On / Reset / Init Sequence

**Scope.** The bring-up ordering an integrator must drive to make the UART operational.

### Init sequence

1. Assert rst\_ni (low) for ≥1 clock; release synchronously. FIFOs clear, registers assume reset values.
2. **Set baud:** set LCR.DLAB=1, program DLL/DLM divisor for the target baud from CLK\_FREQ, clear DLAB.
3. **Set frame format:** program LCR — data bits, parity, stop bits.
4. **Configure FIFOs:** program FCR — enable FIFOs, set RX trigger level, optionally reset TX/RX FIFOs.
5. **Enable interrupts (optional):** program IER for the desired sources; route irq\_o to the interrupt controller.
6. UART is ready — TX via THR, RX via RBR, status via LSR.

### Notes

- All steps are register writes over the peripheral bus (or APB/AXI4-Lite frontend). No special power sequencing beyond the SoC’s standard reset.

## Source cross-refs

- Register detail: [[docs/datasheets/uart/DSHT-10-REGISTER-MAPIDSHT-10]] · RTL: rtl/uart/zen\_uart.sv.

## DSHT-08 · UART · power-on / reset / init.

title: “UART — Programming Model” created: 2026-09-14 status: authored (pending review) standard: shared-agent-infra/DATASHEET-STANDARD.md —

## DSHT-09 · Programming Model

**Scope.** The register-access model and canonical programming sequences. Distills rtl/uart/zen\_uart.sv.

## Access model

- 8 byte-wide registers selected by `addr[4:2]` (3-bit index → word offsets 0x00–0x1C).
- **DLAB** (LCR bit 7) banks RBR/THR and IER onto the divisor latches (DLL/DLM) for baud programming.
- Byte-only data access (`wstrb` unused); read/write via `periph_req_t/periph_rsp_t` or the APB/AXI4-Lite frontend.

## Transmit a byte (polled)

```
poll LSR until THRE == 1      # holding register empty
write THR = data              # DLAB must be 0
```

## Receive a byte (polled)

```
poll LSR until DR == 1        # data ready
read RBR                      # DLAB must be 0
check LSR for OE/PE/FE/BI
```

## Interrupt-driven service

```
on irq_o:
  read IIR                    # decode highest-priority pending cause
  case cause:
    RX-data-available / timeout -> drain RBR to FIFO trigger
    THR-empty                    -> refill THR from SW buffer
    line-status                  -> read LSR, handle error
    modem-status                 -> read MSR
```

## Set baud rate

```
LCR.DLAB = 1
DLL = divisor[7:0]; DLM = divisor[15:8]
LCR.DLAB = 0      # restore RBR/THR/IER access
```

## Source cross-refs

- Register map: [\[\[docs/datasheets/uart/DSHT-10-REGISTER-MAP|DSHT-10\]\]](#) · RTL: `rtl/uart/zen_uart.sv:371,433,483`.

## *DSHT-09 · UART · programming model.*

title: “UART — Register Map” created: 2026-09-14 status: authored (pending review) standard: shared-agent-infra/DATASHEET-STANDARD.md —

## DSHT-10 · Register Map

**Scope.** The CSR map — offsets, access, and function. Offsets are RTL-sourced from the register enum `rtl/include/zen_periph_pkg.sv` (`addr[4:2]` index) and the access logic in `rtl/uart/zen_uart.sv`.

## Register summary

Source: `zen_periph_pkg.sv:66–73` (*index*) — *SOURCED*. Byte offset = index × 4.

| Offset | Index | Name      | Access | DLAB | Function                                   |
|--------|-------|-----------|--------|------|--------------------------------------------|
| 0x00   | 0     | RBR / THR | R / W  | 0    | RX buffer (read) / TX holding (write)      |
| 0x00   | 0     | DLL       | R/W    | 1    | Divisor latch low (baud)                   |
| 0x04   | 1     | IER       | R/W    | 0    | Interrupt enable                           |
| 0x04   | 1     | DLM       | R/W    | 1    | Divisor latch high (baud)                  |
| 0x08   | 2     | IIR / FCR | R / W  | —    | Interrupt ID (read) / FIFO control (write) |
| 0x0C   | 3     | LCR       | R/W    | —    | Line control (frame format, DLAB)          |
| 0x10   | 4     | MCR       | R/W    | —    | Modem control                              |
| 0x14   | 5     | LSR       | R      | —    | Line status                                |
| 0x18   | 6     | MSR       | R/W    | —    | Modem status                               |
| 0x1C   | 7     | SCR       | R/W    | —    | Scratch                                    |

### Key field detail (16550 model)

- **LCR** — [1:0] word length (5–8) · [2] stop bits · [3] parity enable · [4] even parity · [5] stick parity · [6] set-break · **[7] DLAB**.
- **LSR** — [0] DR (data ready) · [1] OE (overflow) · [2] PE (parity) · [3] FE (framing) · [4] BI (break) · [5] THRE · [6] TEMT · [7] RX-FIFO error.
- **IER** — [0] RX-data-available · [1] THR-empty · [2] line-status · [3] modem-status.
- **FCR** — [0] FIFO enable · [1] RX-FIFO reset · [2] TX-FIFO reset · [7:6] RX trigger level.
- **IIR** — [0] pending (0=IRQ active) · [3:1] cause code.

Bit-exact reset values and per-field encodings: **TBC** (owner: verif) where not asserted in rtl/uart/zen\_uart.sv:433–506. Fields listed above are the 16550-standard model implemented by the register read/write logic.

### Source cross-refs

- Index enum: rtl/include/zen\_periph\_pkg.sv:66–73 · Access logic: rtl/uart/zen\_uart.sv:433–506.

### DSHT-10 · UART · register map.

title: “UART — Performance” created: 2026-09-14 status: authored (pending review) standard: shared-agent-infra/DATASHEET-STANDARD.md —

### DSHT-11 · Performance

**Scope.** Throughput and latency envelope. Serial throughput is baud-limited by construction; numbers not fixed by the standard are marked TBC.

### Throughput

| Metric          | Value                                     | Source                                    |
|-----------------|-------------------------------------------|-------------------------------------------|
| Line rate       | = programmed baud (bits/s)                | 16550 model                               |
| Byte rate       | baud ÷ (bits-per-frame)                   | derived                                   |
| Max baud        | CLK_FREQ ÷ (16 × min divisor)             | RTL model; exact value <b>TBC</b> (verif) |
| Back-to-back TX | FIFO-buffered up to FIFO_DEPTH (16) bytes | RTL:18                                    |
| Back-to-back RX | FIFO-buffered up to FIFO_DEPTH (16) bytes | RTL:18                                    |

## Latency

| Metric           | Value                                         | Source      |
|------------------|-----------------------------------------------|-------------|
| Register access  | 1-cycle registered handshake (peripheral bus) | RTL         |
| TX start latency | ≤1 frame after THR write when idle            | TBC (verif) |
| IRQ assertion    | next cycle after qualifying event             | TBC (verif) |

## Notes

- The UART is not a throughput bottleneck at any practical SoC clock; sizing is about FIFO depth vs. interrupt-service latency. Increase FIFO\_DEPTH to relax ISR deadlines.

## Source cross-refs

- RTL: rtl/uart/zen\_uart.sv:18,20 · Verification envelope: [[docs/datasheets/uart/DSHT-14-VERIFICATION-DELIVERABLESIDSH-14]].

## DSHT-11 · UART · performance.

title: “UART — Physical / Implementation” created: 2026-09-14 status: outline (numbers TBC — not yet characterized) standard: shared-agent-infra/DATASHEET-STANDARD.md —

# DSHT-12 · Physical / Implementation

**Scope.** Area, fmax, and constraint pointers. These are process-/flow-dependent and **not yet characterized** — all values TBC pending a PPA run (owner: PD).

## Implementation envelope

| Metric               | Value | Source    |
|----------------------|-------|-----------|
| Cell area            | TBC   | owner: PD |
| Max frequency (fmax) | TBC   | owner: PD |
| Target process       | TBC   | owner: PD |
| Gate count (equiv.)  | TBC   | owner: PD |

## Synthesis notes

- Struct-typed, no SystemVerilog interfaces; passes the mandatory Yosys §11 synth-check (t:\$latch = 0) — no inferred latches. See [[docs/kits/UART-LICENSE-KIT|kit]] compliance statement.
- Single clock domain; only the RX input synchronizer crosses domains.

## Source cross-refs

- RTL: rtl/uart/zen\_uart.sv · Constraints: TBC.

## DSHT-12 · UART · physical / implementation.

## DSHT-13 · RAS & Data Integrity

**Scope.** Error detection and reporting the UART provides. Distills rtl/uart/zen\_uart.sv (LSR).

### Error detection & reporting

| Condition     | Detection                   | Report | Source    |
|---------------|-----------------------------|--------|-----------|
| RX overrun    | FIFO full on new byte       | LSR.OE | RTL (LSR) |
| Parity error  | computed vs received parity | LSR.PE | RTL (LSR) |
| Framing error | invalid stop bit            | LSR.FE | RTL (LSR) |
| Break         | line held low > frame       | LSR.BI | RTL (LSR) |
| RX-FIFO error | any of PE/FE/BI in FIFO     | LSR[7] | RTL (LSR) |

### Reporting path

- Line-status conditions can raise the aggregated irq\_o when enabled (IER.line-status); software reads LSR to classify.
- No ECC/parity on internal FIFOs (not required at this altitude; data-integrity offload available via [[docs/catalog/IP-CATALOG-09-DATA-INTEGRITYICRC]] if needed at the protocol layer).

### Source cross-refs

- RTL: rtl/uart/zen\_uart.sv (LSR assembly, IIR) · Registers: [[docs/datasheets/uart/DSHT-10-REGISTER-MAPIDSHT-10]].

### DSHT-13 · UART · RAS & data integrity.

## DSHT-14 · Verification & Deliverables

**Scope.** The verification environment and the deliverables manifest. Distills the [[docs/kits/UART-LICENSE-KITUART license kit]].

### Verification

| Item               | Value                                        | Source        |
|--------------------|----------------------------------------------|---------------|
| Methodology        | assertion-based cocotb, byte-exact checks    | kit           |
| Test count         | 10 / 10 passing                              | kit (SOURCED) |
| Gates              | Verilator DV + Yosys §11 synth (t:\$dlatc=0) | kit           |
| Shared VIP         | periph_codecs.py bus codecs                  | kit           |
| Independent re-run | re-verified by Arch Lead in review           | kit           |

### Deliverables manifest (kit)

- Parameterized RTL: rtl/uart/zen\_uart.sv + APB/AXI4-Lite frontends.
- Compliance testbenches + shared VIP.
- Integration docs (this datasheet + [[docs/datasheets/uart/DSHT-15-INTEGRATION-GUIDEIDSHT-15]]).
- Compliance statement (two-gate evidence).

### Source cross-refs

- Kit: [[docs/kits/UART-LICENSE-KITUART-LICENSE-KIT]] · Regression: scripts/run-regression.sh.

DSHT-14 · UART · verification & deliverables.

title: "UART — Integration Guide" created: 2026-09-14 status: authored (pending review) standard: shared-agent-infra/DATASHEET-STANDARD.md —

DSHT-15 · Integration Guide

**Scope.** Instantiation, tie-offs, and DFT pointers for the integrator. Distills rtl/uart/.

Instantiation

- Pick the frontend for your fabric: native zen\_uart (periph bus), zen\_uart\_apb (APB4), or zen\_uart\_axil (AXI4-Lite).
- Set parameters: FIFO\_DEPTH (default 16), CLK\_FREQ (Hz, for documentation/ baud aid).

Connections

| Signal                              | Connect to                                                                |
|-------------------------------------|---------------------------------------------------------------------------|
| clk_i / rst_ni                      | SoC clock / active-low reset                                              |
| bus port (periph / APB / AXI4-Lite) | host fabric via [[docs/catalog/IP-CATALOG-01-BUS-FABRIC Bus Fabric]]      |
| uart_tx_o / uart_rx_i               | off-chip serial transceiver / pads                                        |
| irq_o                               | interrupt controller ([[docs/catalog/IP-CATALOG-02-INTERRUPTS PLIC/AIA]]) |

Tie-offs & notes

- uart\_rx\_i may be asynchronous — internal 2-FF sync handles metastability; no external synchronizer needed.
- Unused modem signals are register-modeled (no external pins required for basic 3-wire operation: TX, RX, GND).

DFT

- Single clock domain simplifies scan insertion; reset is active-low async. Scan/DFT hooks: TBC (owner: DFT).

Source cross-refs

- RTL: rtl/uart/zen\_uart.sv, zen\_uart\_apb.sv, zen\_uart\_axil.sv · Bridges: [[docs/BRIDGE-INTEGRATION-GUIDE|bridge integration guide]].

DSHT-15 · UART · integration guide.

title: "UART — Ordering / Licensing / Support" created: 2026-09-14 status: authored (pending review) standard: shared-agent-infra/DATASHEET-STANDARD.md —

DSHT-16 · Ordering / Licensing / Support

**Scope.** Commercial terms. Distills the [[docs/kits/UART-LICENSE-KIT|UART license kit]] and the portfolio licensing model.

Product

| Item          | Value                                                                             |
|---------------|-----------------------------------------------------------------------------------|
| Kit           | zen-uart-kit-v2                                                                   |
| Catalog entry | [[docs/catalog/IP-CATALOG-04-SERIAL-IO IP-CATALOG vol. 04 · Serial I/O]]          |
| Deliverable   | parameterized RTL + VIP + compliance TB + integration docs + compliance statement |

## Licensing

- **Internal soft-license** by default (free IP flow within zen-semi); kit is license-ready-structured for external.
- Packaged via `scripts/package-kit.sh`; kit-eligible only after passing both gates (Verilator DV + Yosys §11).

## Support

- Owner: zen-peripheral (RHA: Amanda Chen). Issues via the standard tracker; integration questions via the kit contact.

## Source cross-refs

- Kit: [[docs/kits/UART-LICENSE-KIT|UART-LICENSE-KIT]] · Licensing model: [[docs/catalog/IP-CATALOG-00-INDEX|catalog index]].

## DSHT-16 · UART · ordering / licensing / support.

title: “UART — Appendices” created: 2026-09-14 status: authored (pending review) standard: shared-agent-infra/DATASHEET-STANDARD.md —

## DSHT-A · Appendices

Scope. Full signal descriptions, acronyms, references, and revision history.

## Appendix A — Signal descriptions (full port table)

Source: `rtl/uart/zen_uart.sv` module header — SOURCED.

| Signal    | Dir | Type         | Description                                                    |
|-----------|-----|--------------|----------------------------------------------------------------|
| clk_i     | in  | logic        | Reference clock.                                               |
| rst_ni    | in  | logic        | Active-low asynchronous reset.                                 |
| req_i     | in  | periph_req_t | Peripheral-bus request (addr/wdata/write/valid; wstrb unused). |
| rsp_o     | out | periph_rsp_t | Peripheral-bus response (rdata/ready).                         |
| uart_tx_o | out | logic        | Serial transmit output.                                        |
| uart_rx_i | in  | logic        | Serial receive input (internally synchronized).                |
| irq_o     | out | logic        | Aggregated interrupt (IIR-decoded).                            |

## Appendix B — Acronyms

- **RBR/THR** receive-buffer / transmit-holding · **IER** interrupt-enable · **IIR/FCR** interrupt-ID / FIFO-control · **LCR** line-control · **MCR/MSR** modem-control/status · **LSR** line-status · **SCR** scratch · **DLAB** divisor-latch-access-bit · **DLL/DLM** divisor-latch low/high.

## Appendix C — References

- shared-agent-infra/DATASHEET-STANDARD.md (this standard) · NS16550A functional spec (de-facto) · ARM APB4 / AXI4-Lite.
- Source: `rtl/uart/zen_uart.sv`, `zen_uart_apb.sv`, `zen_uart_axil.sv`, `rtl/include/zen_periph_pkg.sv`.

## Appendix D — Revision history

| Date       | Rev | Change                                                | Author            |
|------------|-----|-------------------------------------------------------|-------------------|
| 2026-09-14 | 0.1 | Initial datasheet authored from as-built RTL (pilot). | Amanda Chen (RHA) |

*DSHT-A · UART · appendices.*