June 2022

On-Chip Double Data Rate Serial Bus

n/a

On-Chip Double Data Rate Serial Bus

ABSTRACT

Silicon chip area is expensive, especially for cutting-edge process nodes. Moreover, on-chip wire density has not been scaling well with technology. This disclosure describes techniques that leverage n:1 serdes designs, e.g., double data rate (DDR) design, to reduce wire count and congestion, thus freeing up scarce silicon area. For example, if n=2 (double data rate), wire count and congestion can be reduced by half.

KEYWORDS

- Double data rate (DDR)
- Quadruple data rate
- System on chip (SoC)
- On-chip serdes
- Serdes
- Scan chain
- Edge triggered flip-flop
- Single data rate
- Design for testability (DFT)

BACKGROUND

Silicon chip area is expensive, especially for cutting edge process nodes. Moreover, on-chip wire density has not been scaling well with technology. While on-chip wire densities were perhaps acceptable at the 7 nm process node, they become too high at the 5 nm node and beyond. As process nodes advance, conventional designs for on-chip bit transport increasingly slip performance, scale, and total cost of ownership (TCO) targets.

Single data rate (SDR) design (with rising or falling sampling) is widely used for on-chip transport design. As the wire density saturates, SDR can become highly area inefficient.

DESCRIPTION

This disclosure describes techniques that leverage n:1 serdes (serialization/deserialization) designs, e.g., double data rate (DDR) design, to create a standard-cell-like macro that substantially reduces on-chip wire count and congestion and frees up scarce silicon area. For example, if n=2, corresponding to DDR, wire count and congestion can be reduced by half. If n=4, corresponding to quadruple data rate, wire count and congestion can be reduced to a quarter.

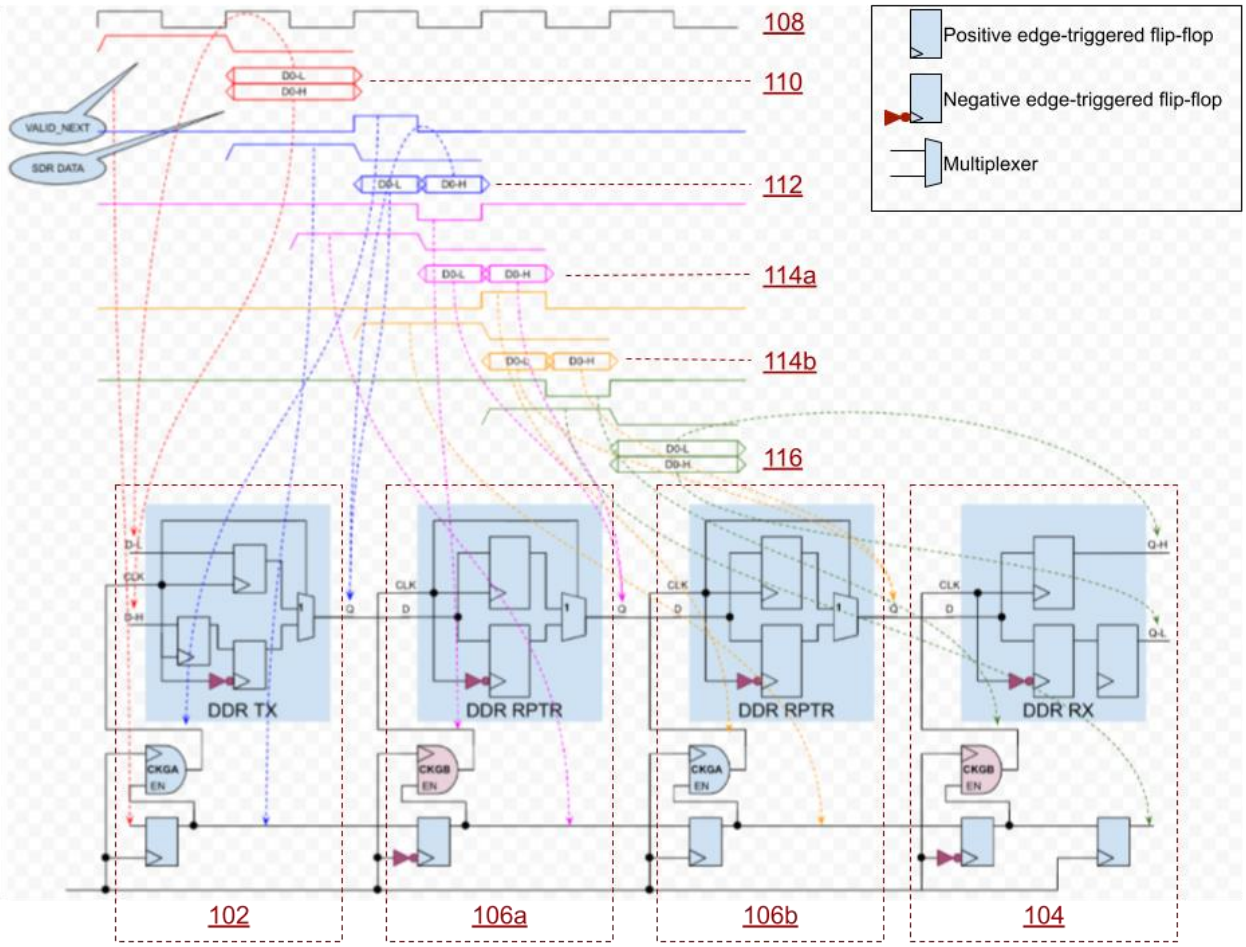

Fig. 1: On-chip double data rate serial bus

Fig. 1 illustrates an example on-chip DDR serial bus. Data is to be transmitted between an on-chip transmitter (102) and an on-chip receiver (104). Between the transmitter and receiver,

there can be zero or more repeaters (106a-b), whose function is to maintain signal integrity, e.g., to ensure that the signal maintains noise margin specifications as it travels from the transmitter to the receiver.

The serial bus is driven by a clock (108). Two bits of data, D0-H and D0-L (110), are presented each clock cycle at the input to the transmitter, such that both bits parallelly occupy a full clock cycle. The transmitter transforms the parallel bits of data to serial, e.g., time-multiplexed, data (112), such that the output of the transmitter requires only one wire (Q) as compared to its input (D-H and D-L), which requires two.

The serialized, single-wire bits 112 each occupy half a clock cycle. The single-wire bits travel through the repeaters and reach the receiver, maintaining, as they travel, their serial nature and their half-clock occupancy (114a-b). At the receiver 104, the single-wire bits are deserialized to revert to two-wire transport (one bit at Q-H and one at Q-L) and parallel occupancy of a full clock cycle (116). In contrast to conventional SDR design, which requires two wires to pass two bits of data from the transmitter through the repeaters to the receiver, the described techniques require only a single wire from the output of the transmitter to the input of the receiver.

The techniques apply to n:1 serdes, where n is the number of parallel bits being serialized per clock cycle. For example, n=2 corresponds to double data rate; n=4 corresponds to quadruple data rate (where four parallel bits, requiring four wires, are serialized into four serial bits requiring one wire); etc. The techniques apply to various transistor logic families, e.g., full-swing CMOS; low-swing CMOS; differential signals; multi-level (non-binary) digital signals; etc. The techniques can be synchronous, e.g., the flip-flops can share a common clock distributed by a clock tree. Alternatively, the techniques can be based on clock-forwarding, e.g., the

transmitter forwards its clock to the repeater, which forwards its clock to the next repeater, and so on until the last repeater, which forwards its clock to the receiver.

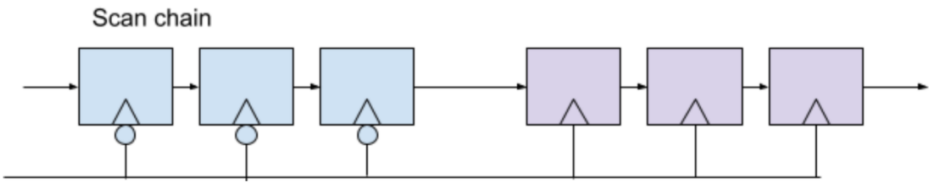

Fig. 2: Scan chain connecting falling edge-triggered flip-flops followed by rising edge-triggered flip-flops

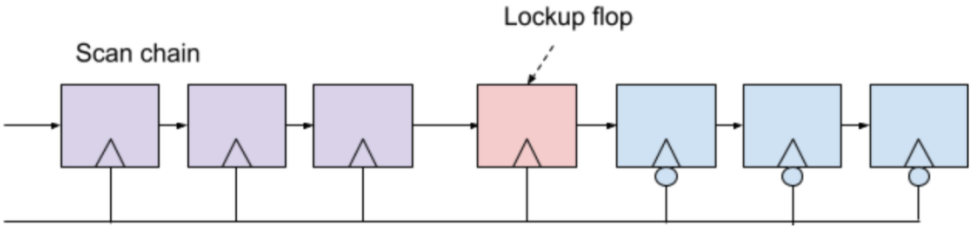

Fig. 3: Scan chain connecting rising edge-triggered flip-flops followed by falling edge-triggered flip-flops, via a lock-up flip-flop

A feature of the described on-chip serial bus is that it is compatible with the design for testability (DFT) paradigm and with automatic test pattern generation (ATPG). For example, as illustrated in Fig. 2, a scan chain can be formed by connecting a chain of negative (falling, indicated in blue) edge-triggered flip-flops (of Fig. 1) to a chain of positive (rising, mauve) edge-triggered flip-flops. Alternatively, as illustrated in Fig. 3, a scan chain can be formed by connecting a chain of rising edge-triggered flip-flops to a chain of falling edge-triggered flip-flops via a lockup flip-flop.

Both alternatives enable the system-on-chip to have just one scan-in pin and one scan-out pin, thereby optimizing pin count while enabling visibility of flip-flop content during testing via serialized scanning in and scanning out. Specifically, the techniques obviate separate scan chains (and corresponding multiple scan-in/out pins) for negative edge-triggered and positive edge-triggered flip-flops.

CONCLUSION

This disclosure describes techniques that leverage n:1 serdes designs, e.g., double data rate (DDR) design, to reduce wire count and congestion, thus freeing up scarce silicon area. For example, if n=2 (double data rate), wire count and congestion can be reduced by half.