Vivado Utilization Report: hello_world_arty_a7 Design

Tool Version: Vivado v.2018.3 (win64)
Build: 2405991 Thu Dec 6 23:38:27 MST 2018
Date: Thu Oct 17 09:45:18 2024
Host: Samuel, 64-bit OS (build 9200)
Command: report utilization -file hello_world_arty_a7_utilization_synth.rpt -pb hello_world_arty_a7_utilization_synth. pb
Design: hello_world_arty_a7
Device: 7a100ticsg324-1L
Design State: Synthesized

Utilization Design Information

This report provides an overview of the resource utilization of the design after synthesis. It includes details on slice logic, memory, DSP, I/O, clocking, and other specific features. Below is the breakdown:

---

1. Slice Logic

Site Type	Used	Fixed	Available	Util%
Slice LUTs	2400	0	63400	3.79%
LUT as Logic	2400	0	63400	3.79%
LUT as Memory	0	0	19000	0.00%
Slice Registers	1768	0	126800	1.39%
F7 Muxes	339	0	31700	1.07%
F8 Muxes	0	0	15850	0.00%

Note: The final LUT count is typically lower after physical optimizations. Running opt design can provide a more realistic count.

1.1. Summary of Registers by Type

Total	Clock Enable	Synchronous	Asynchronous
0	_	-	-
0	_	-	Set
0	_	-	Reset
0	_	Set	-
88	Yes	Set	-
1680	Yes	Reset	-

---

2. Memory

Site Type	Used	Fixed	Available	Util%
Block RAM	2	0	135	1.48%
RAMB36/FIFO	2	0	135	1.48%
RAMB18	0	0	270	0.00%

---

3. DSP

Site Type	Used	Fixed	Available	Util%
DSPs	0	0	240	0.00%

---

4. IO and GT Specific

Site Type	Used	Fixed	Available	Util%
Bonded IOB	3	0	210	1.43%
Bonded IPADs	0	0	2	0.00%

---

5. Clocking

Site Type	Used	Fixed	Available	Util%
BUFGCTRL	2	0	32	6.25%
MMCME2_ADV	0	0	6	0.00%
PLLE2_ADV	0	0	6	0.00%

6. Specific Features

Site Type	Used	Fixed	Available	Util%
BSCANE2	0	0	4	0.00%
STARTUPE2	0	0	1	0.00%

7. Primitives

Primitive	Used	Functional Category
FDRE	1680	Flop & Latch
LUT6	1270	LUT
MUXF7	339	MuxFx
CARRY4	158	Carry Logic

8. Black Boxes

None were used in this design.

9. Instantiated Netlists

None were instantiated in this design.

Synthesis Report for RVSteel MCU Instances

Overview

This report details the synthesis and optimization process for the RVSteel MCU design, focusing on various key aspects such as logic optimization, instance merging, RAM mapping, timing optimization, and technology mapping. Throughout the synthesis process, the primary goal was to optimize resource usage, ensure timing constraints were met, and improve the overall efficiency of the design.

Optimization and Instance Merging

During the synthesis, significant efforts were made to optimize the design by merging sequential elements and propagating constants across instances. Sequential elements such as rx_reg_reg, rx_data_reg, and other related registers from rvsteel_spi_instance and rvsteel_uart_instance were combined to reduce redundancy and minimize logic complexity. This merging resulted in an overall reduction of the design size, which improves both performance and resource utilization.

Key instances involved in this process included:

- **SPI and UART Modules:** The merging of sequential elements from these modules led to a more streamlined architecture, with registers like rx_data_reg[5], rx_data_reg[4], rx_reg_reg[1], and rx_reg_reg[3] being simplified. This reduced the number of logic gates needed and allowed for more efficient synthesis of the overall design.
- **Constant Propagation:** Constants were propagated across these sequential elements to further optimize the circuit, eliminating unnecessary gates and logic checks. This step improved not only the resource efficiency but also the timing performance of the design.

RAM Mapping and Resource Allocation

The RAM blocks (ram_reg_0 and ram_reg_1) were a critical part of the design. These were implemented using Block RAM, with each RAM block being configured as 2 K x 32 for both read and write operations. The mapping was performed under the following configurations:

- **Preliminary Block RAM Mapping:**
  - ram_reg_0 and ram_reg_1: Each configured as 2 K x 32, with the Read configuration set to **READ_FIRST** and the Write configuration set to **WRITE_FIRST**.

- **Final Block RAM Mapping:**
  - The final mapping remained consistent with the preliminary configuration. This ensures that the memory was optimally utilized while maintaining consistency between read and write operations.

The RAM usage was a crucial factor in maintaining the design's overall performance, with the consistent mapping across all stages contributing to a reliable synthesis result.

Timing Optimization and Constraints

During synthesis, timing constraints were a major focus, particularly in the context of RAM blocks and other critical paths in the design. The synthesis report indicated the following:

- **Block RAM Timing Issues:** A recurrent issue observed throughout the synthesis runs was the sub-optimal timing of instances such as ram_reg_0 and ram_reg_1. This was primarily due to the **absence of optional output registers** on the Block RAMs, which led to slight delays in signal propagation. Adding output registers could potentially address this issue and improve timing performance.
- **Timing Optimization Efforts:** Despite the timing limitations of the RAM blocks, the synthesis tool successfully applied timing constraints across the rest of the design. The overall timing was optimized for instances that didn't rely on Block RAM, ensuring that the critical paths outside of these memory elements were handled efficiently.

Technology Mapping and Resource Usage

The technology mapping process successfully translated the optimized logic into a hardware implementation that minimized resource usage. Key highlights of resource utilization include:

- **Look-Up Tables (LUTs):** Various levels of LUTs (LUT1 to LUT6) were used throughout the design to implement combinatorial logic. The synthesis process aimed to minimize the number of LUTs while maximizing their utilization, which contributed to an overall reduction in the design's complexity.
- **Flip-Flop-based Cells (FDRE/FDSE):** Flip-flop-based cells were employed for data storage and sequential logic, providing a balance between speed and resource usage.
- **Block RAM (RAMB36):** As mentioned earlier, Block RAM resources were utilized for the memory elements in the design. The consistent 2 K x 32 configuration ensured optimal memory usage across the different modules.

Final Synthesis Summary

The synthesis of the RVSteel MCU instances was completed successfully with no errors or warnings reported. The final netlist preparation and optimization efforts yielded a design that efficiently uses the available resources while adhering to the defined timing constraints. The primary areas of improvement that could be explored in future iterations include:

- **Adding Optional Output Registers:** This would help mitigate the timing delays in the Block RAM, particularly for instances like ram_reg_0 and ram_reg_1.
- **Further Logic Compression:** While the current optimization process has reduced logic complexity significantly, there may be additional opportunities for further compression, especially in non-memory-related components of the design.

Conclusion

The synthesis process for the RVSteel MCU design has resulted in a well-optimized, resource-efficient hardware implementation. Constant propagation, instance merging, and careful RAM mapping played key roles in achieving this result. While minor timing inefficiencies were observed due to Block RAM output registers, the overall design is robust and well-suited for the intended application.