SHA-256 V2

IMPLEMENTATION DETAILS

(25 Aug 2025)

**Overview**

This repository contains a Verilog implementation of the SHA-256 cryptographic hash algorithm in single-block mode (512-bit input → 256-bit digest). The project is designed for FPGA targets (tested on Xilinx Artix-7 XC7A200T) and verified using QuestaSim 2024.1.

**Toolchain & Environment**

* **Vivado**: 2024.1.2 (64-bit)
* **Simulator**: QuestaSim 2024.1 (Feb 2024 build)
* **Hardware (simulated)**: Artix-7 XC7A200T, 100–150 MHz
* **Host**: Windows 11 Pro

**The implementation includes:**

* Modularized RTL design (sha256\_core, FSM controller, K-constants memory, and top wrapper).
* Testbench for simulation and functional verification.
* Waveform results demonstrates correct operation.
* Documentation of design choices and test methodology.

**Features**

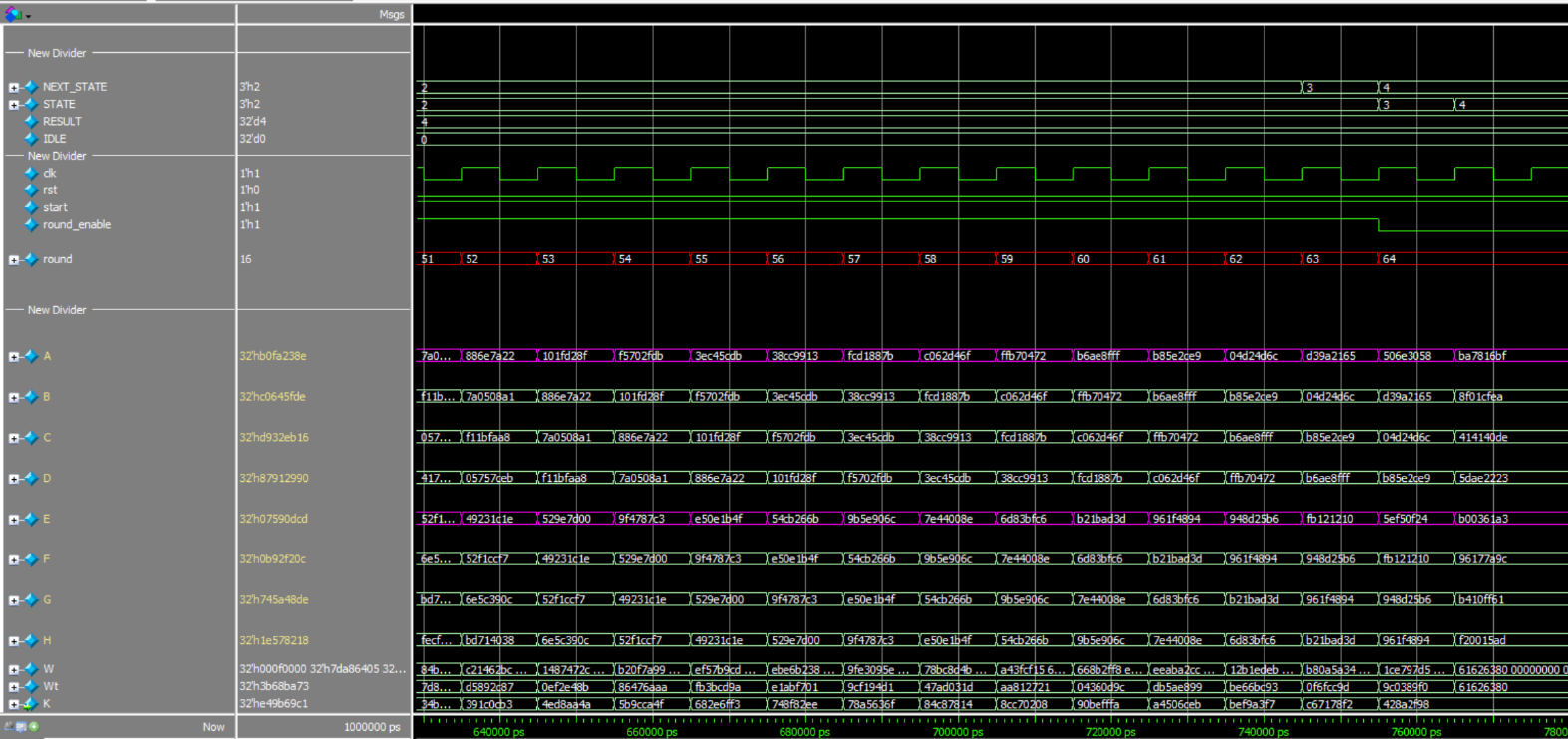
* **Algorithm**: SHA-256 (FIPS 180-4 compliant)
* **Mode**: Single-block (Example B.1 from FIPS 180-2)
* **Digest Size**: 256 bits
* **Design Language**: Verilog-2001
* **Simulation Tool**: QuestaSim 2024.1
* **Target Device**: Xilinx Artix-7 XC7A200T (XC7A200T-SBG484-1)
* **Operating Frequency**: 100–150 MHz (simulation validated)

**File Structure**

* FSM.v Finite State Machine for round and control logic
* K\_mem.v ROM containing SHA-256 round constants (K[0..63])
* SHA\_TOP.v Top-level wrapper connecting FSM, core, and memory
* sha256\_core.v Core compression function (working variables A–H, Wt, etc.)
* TB.v Testbench with input vectors and simulation control
* results.png Sample simulation waveform
* SHA256.docx Implementation document

**Module Descriptions**

* **FSM.v**
  + Implements the finite state machine to control SHA-256 operations.
  + Handles IDLE, START, ROUND, and RESULT states.
  + Controls iteration over 64 rounds.
  + Generates round\_enable and synchronization signals.
* **K\_mem.v**
  + Defines the SHA-256 round constants (K[0..63]) as a ROM.
  + Each constant is 32-bit.
  + Indexed by the current round number.
* **sha256\_core.v**
  + Implements the SHA-256 compression function:
  + Eight working registers (A–H).
  + Message schedule expansion (Wt).
  + Core functions (Ch, Maj, Σ0, Σ1, σ0, σ1).
  + Updates state values for each round.
* **SHA\_TOP.v**
  + Top-level module integrating that generates 512 bits input block
  + FSM controller.
  + Constant memory (K\_mem).
  + Core function (sha256\_core).
  + Provides clean I/O interface for start/reset/data input and digest output.
* **TB.v (Testbench)**
  + Drives clock/reset/start signals.
  + Observes output digest and round activity.
  + Contains $monitor and $stop for debugging.
* **Simulation Results**
* **Waveform (SHA256\_results.png)** shows state transitions, working variable updates (A–H), and round outputs.



* Digest stabilizes after **64 rounds**, matching expected SHA-256 values.