FPGA Implementation of the SHA-256 Algorithm

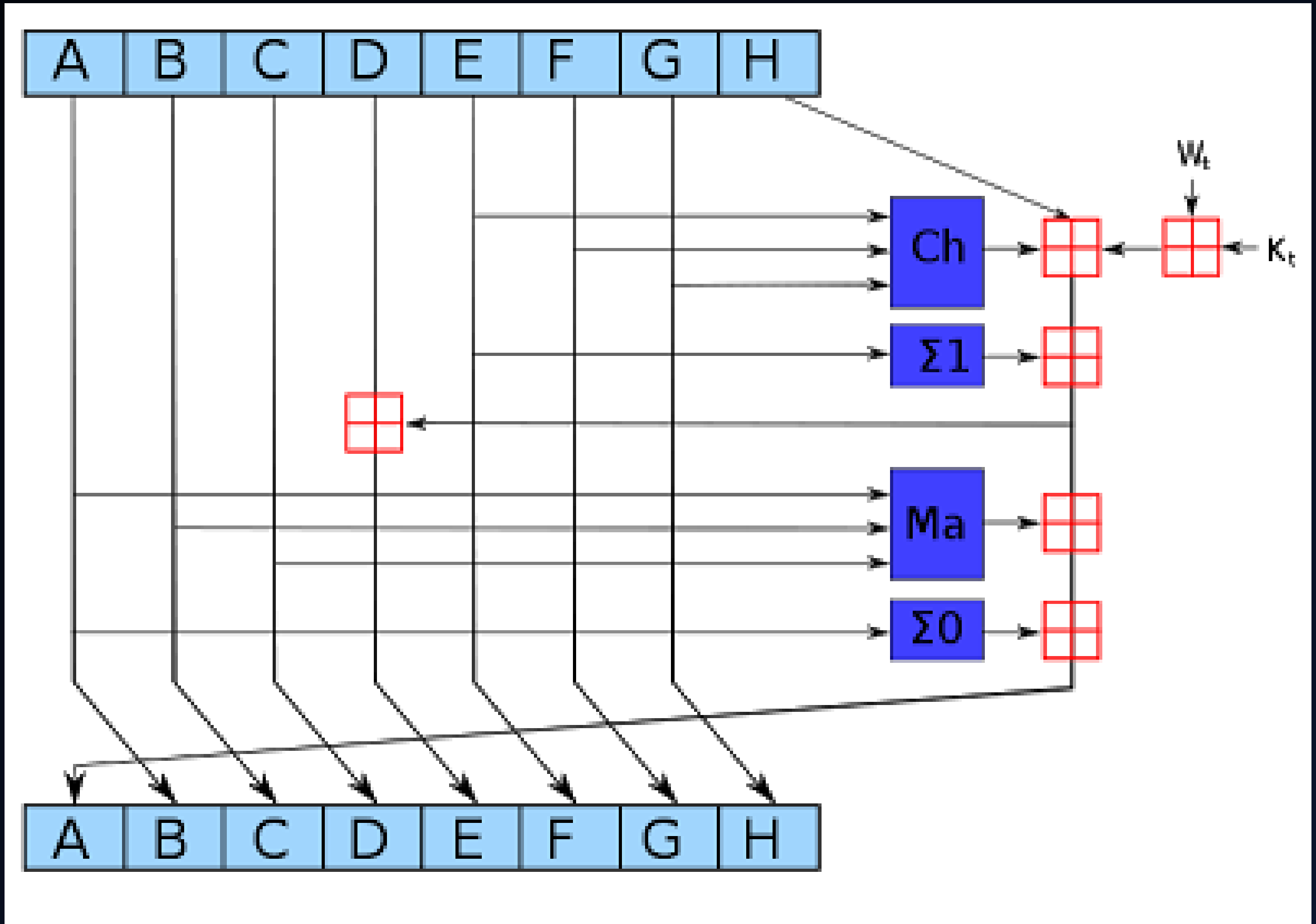
DIGITAL SYSTEMS II – FINAL PROJECT PROPOSAL

Jeremy Maxey-Vesperman & Zach Butler

SHA-wha???

- 256-bit Secure Hash Algorithm 2 (SHA-256)
- Designed by the U.S. National Security Agency
- Widely used in security protocols
  - TLS, SSL, etc.
- More recently implemented in cryptocurrency
  - Bitcoin

The Algorithm



Bit…coin?

- Decentralized digital cryptocurrency
- Based on blockchain technology
- Exponential increase in users and value as a fiat currency
- Transactions secured using double-SHA-256

Bitcoin mining

- Record-keeping service performed by computers in the network
- Uses proof-of-work system to accept new group of transactions
  - Must find a value (nonce) that, when hashed with block content, is below the target difficulty level
- Requires brute force tactic to find an acceptable nonce
  - Current average number of nonce trials required = 6x10^21 trials

How does this tie together?

- Mining difficulty auto-adjusts to keep block creation to ~10min.
- Probability of discovering an acceptable nonce is incredibly low with CPUs/GPUs
- Much greater chance with an ASIC optimized for SHA-256 hashing
- Can get close to ASIC performance using an FPGA

The Goal

- Use Verilog to implement SHA-256 on Altera DE2 Board
- Input String: I LOVE DIGITAL SYSTEMS II!
- SHA-256 Hash: 406c679ff665639e638db762c9ffdcb6e6616b4f7a223152ab86de0221e4a3af

Questions?