DIGITAL SYSTEMS DESIGNS FINAL PROJECT
COMPUTER ENGINEERING

HUMMINGBIRD LIGHTWEIGHT CRYPTO-CORE

ABIDZAR RABBANI KHATIB HARAHAP 2406369034
NAUFAL RAFIF ADIGHAMA 2406368965
ZIYADZHARIF ALFARABI KURNIAWAN 2406369053
FAHREZA ARSYA MAULANA 2406450365

BACKGROUND

THE CONSUMPTION TENDENCIES AND PURCHASING ACTIVITIES OF GENERATION Z WITH RESPECT TO SUSTAINABLE FASHION ARE CHANGING, AND HOW YOU WANT TO DEFINE THAT WILL BE IMPORTANT FOR HOW YOU APPROACH THIS REPORT. BECAUSE OF BEING THE FIRST DIGITALLY NATIVE GENERATION, GENERATION Z IS BECOMING ECONOMICALLY POWERFUL, HAS A STRONG COMMITMENT TO ENVIRONMENTALLY SUSTAINABLE PRACTICES, AND HAS THE POTENTIAL TO INFLUENCE HOW RETAIL OPERATES. THIS REPORT PULLS TOGETHER VARIOUS RESEARCH STUDIES ON WHAT MOTIVATES GENERATION Z TO MAKE SUSTAINABLE CHOICES, INCLUDING EXPECTATIONS FOR TRANSPARENCY, AUTHENTICITY, AND SOCIAL RESPONSIBILITY FROM THE BRANDS THEY CHOOSE TO SUPPORT. IN ADDITION, THIS REPORT ALSO DISCUSSES GENERATION Z'S PURCHASE BEHAVIOUR INCONSISTENCIES, OR THE VALUE-ACTION GAP, WHICH REFERS TO THE DIFFERENCE IN PRO-SUSTAINABILITY ATTITUDES AND ACTUAL PURCHASE BEHAVIOURS, LARGELY CAUSED BY FINANCIAL LIMITATIONS AND THE IMPACT OF FAST FASHION ON GENERATION Z'S BEHAVIOUR.

PROJECT DESCRIPTION

IN TERMS OF A "PROJECT," THIS IS THE DEVELOPMENT OF AN HUMMINGBIRD CRYPTO-SYSTEM THAT ALLOWS LIGHTWEIGHT CRYPTO SERVICES THROUGH THE USE OF THE VHDL PROGRAMMING LANGUAGE. THE HUMMINGBIRD ENCRYPTS AND DECRYPTS DATA BLOCKS 16 BITS WIDE, WITH EACH BLOCK USING A 256-BIT KEY, BY MEANS OF THE FOLLOWING COMPONENTS: S-BOX SUBSTITUTION TABLES, LINEAR DIFFUSION TRANSFORMATION (THE LFT), AND SEVERAL INTERNAL REGISTERS. THE FIRST INTERNAL REGISTER IS UPDATED BY A GALOIS-BASED LINEAR FEEDBACK SHIFT REGISTER OR LFSR, CREATING THE PSEUDO-RANDOMNESS FOR THE OUTPUT. ALL FUNCTIONS OF THE SYSTEM ARE ORGANIZED BY FUNCTIONAL MODULES: KEY LOADING, STATE INITIALIZATION, ROUNDS OF ENCRYPTION, AND STATE UPDATING. A CONTROL MODULE CONTROLS AND MONITORS THE OPERATION OF THE HUMMINGBIRD SYSTEM USING EITHER A FINITE STATE MACHINE (FSM) OR A MICRO-PROGRAMMED SEQUENCER. THE HUMMINGBIRD DESIGN METHODOLOGY CONTRIBUTES TO THE OVERALL PERFORMANCE OPTIMIZATION OF THE FPGA THROUGH THE USE OF A COMBINATION OF DATAFLOW, BEHAVIORAL, AND STRUCTURAL MODELING OF THE HUMMINGBIRD SYSTEM WITHIN THE FPGA ARCHITECTURAL STRUCTURE OF THE DESIGN.

OBJECTIVES

TO DESIGN AND IMPLEMENT A COMPLETE LIGHTWEIGHT CRYPTOGRAPHIC CORE BASED ON THE HUMMINGBIRD ALGORITHM USING VHDL.

TO REALIZE A FUNCTIONAL ENCRYPTION AND DECRYPTION ENGINE THAT OPERATES ON 16-BIT DATA WITH A 256-BIT KEY AND UTILIZES INTERNAL STATE REGISTERS AND A 16-BIT GALOIS LFSR.

TO EVALUATE THE SUITABILITY OF THIS DESIGN FOR FPGA IMPLEMENTATION IN TERMS OF ARCHITECTURE, MODULARITY, AND POTENTIAL FOR LOW-AREA AND LOW-POWER DEPLOYMENT IN REAL EMBEDDED SYSTEMS.

TO DEMONSTRATE THE USE OF MULTIPLE VHDL MODELING STYLES (DATAFLOW, BEHAVIORAL, STRUCTURAL, FSM, FUNCTIONS, AND PROCEDURES) IN A SINGLE, COHERENT HARDWARE DESIGN.

TO VERIFY CORRECT OPERATION OF THE CORE THROUGH SIMULATION, SHOWING PROPER INITIALIZATION, ROUND-BASED ENCRYPTION, AND SUCCESSFUL RECOVERY OF THE PLAINTEXT AFTER DECRYPTION.

USED TOOLS

- GITHUB
- VIVADO

IMPLEMENTATION

ABIDZAR'S PART

ABIDZAR WAS RESPONSIBLE FOR THE CORE COMBINATIONAL LOGIC THAT REPRESENTS THE MATHEMATICAL BACKBONE OF THE HUMMINGBIRD CIPHER. EVEN THOUGH OUR PROTOTYPE USED SIMPLIFIED PLACEHOLDER OPERATIONS, THE STRUCTURE MIRRORS WHAT A REAL LIGHTWEIGHT CIPHER REQUIRES: SMALL LOOKUP TABLES, FIXED TRANSFORMS, AND DETERMINISTIC MIXING OPERATIONS.

THIS PART OF THE SYSTEM DOES NOT STORE ANYTHING; IT SIMPLY TRANSFORMS 16-BIT VALUES EVERY CYCLE. THE DATAPATH MODULES CREATED HERE SERVE AS THE CRYPTOGRAPHIC "WORK UNITS" THAT THE CONTROLLER ACTIVATES AT DIFFERENT STAGES. BY COMPLETING THIS PORTION, WE ENSURED THAT THE DESIGN HAD A FUNCTIONAL ARITHMETIC AND SUBSTITUTION LAYER THAT COULD BE REUSED IN EVERY ENCRYPTION ROUND.

```
SBOX_OUT <= SBOX(TO_INTEGER(UNSIGNED(DATA_IN)));
LT_OUT <= DATA_IN XOR ROTATE_LEFT(DATA_IN, 3);
MIX_OUT <= LT_OUT XOR KEY_IN;
```

REZA'S PART

REZA DEVELOPED THE SEQUENTIAL BACKBONE OF THE SYSTEM, INCLUDING THE 16-BIT LFSR AND THE FOUR 16-BIT STATE REGISTERS (RS1–RS4). HIS MODULES RESPOND TO THE GLOBAL CLOCK AND THE WRITE-ENABLE SIGNALS COMING FROM THE CONTROLLER, MAKING THEM RESPONSIBLE FOR STORING INTERMEDIATE STATES ACROSS MULTIPLE CYCLES. HIS LFSR LOGIC ALLOWS THE SYSTEM TO GENERATE PSEUDO-RANDOM VALUES REQUIRED FOR LIGHTWEIGHT CIPHERS, WHILE THE REGISTER BANK MAINTAINS THE EVOLVING ENCRYPTION STATE.

```
IF RISING_EDGE(CLK) THEN
  IF RST='1' THEN
    LFSR_REG <= (OTHERS=>'0');
  ELSIF LOAD='1' THEN
    LFSR_REG <= SEED;
  ELSIF ENABLE='1' THEN
    LFSR_REG <= LFSR_NEXT_VALUE;
  END IF;
END IF;
```

NAUFAL'S PART

NAUFAL CREATED THE CONTROLCORE FINITE-STATE MACHINE, WHICH ACTS AS THE BRAIN OF THE ENTIRE SYSTEM. HIS CONTROLLER DETERMINES THE OVERALL TIMING, SEQUENCING, AND SYNCHRONIZATION OF EVERY OTHER MODULE. THE FSM TRANSITIONS THROUGH RESET, IDLE, INITIALIZATION, AND FOUR ENCRYPTED ROUND PHASES BEFORE SIGNALING COMPLETION. THE MODULE ASSERTS WRITE-ENABLES, SELECTS ACTIVE BLOCKS, INCREMENTS THE ROUND COUNTER, AND MANAGES THE LFSR'S BEHAVIOR.

NAUFAL'S CONTRIBUTION IS ESSENTIAL FOR ORCHESTRATING THE DATAPATH, ENSURING EACH STAGE OCCURS IN THE CORRECT ORDER AND THAT REGISTERS UPDATE ONLY DURING THEIR INTENDED ROUND. WITHOUT HIS CONTROL LOGIC, THE REST OF THE SYSTEM WOULD NOT OPERATE IN A COORDINATED FASHION.

```
CASE STATE_REG IS
  WHEN S_INIT =>
    LFSR_LOAD_I <= '1';
    RS1_WE_I  <= '1';
    RS2_WE_I  <= '1';
    ...
  WHEN S_ENC_R1 =>
    BLOCK_SEL_I <= "00";
    RS1_WE_I  <= '1';
  WHEN S_ENC_R2 =>
    BLOCK_SEL_I <= "01";
    RS2_WE_I  <= '1';
```

ZIYAD’S PART

ZIYAD HANDLED THE TOP-LEVEL INTEGRATION OF ALL SUBSYSTEMS AND CREATED THE COMPLETE TESTBENCH ENVIRONMENT USED TO VERIFY THE DESIGN. HIS TOP-LEVEL ENTITY INSTANTIATES EVERY MODULE—CONTROLLER, LFSR, REGISTER BANK, AND KEY SCHEDULER—AND CONNECTS THEM TO FORM A FUNCTIONAL CRYPTOGRAPHIC CORE. HE ALSO DEVELOPED THE TESTBENCH SIGNALS, INCLUDING CLOCK GENERATION, RESET, START PULSES, AND INPUT KEYS, ALLOWING THE ENTIRE SYSTEM TO BE SIMULATED IN VIVADO.

```
U_CONTROLCORE : CONTROLCORE PORT MAP( ... );
U_LFSR16   : LFSR16   PORT MAP( ... );
U_STATEREGS : STATEREGISTERBANK PORT MAP( ... );
U_KEYSCHED  : KEYSCHEDULER PORT MAP( ... );
```

OVERALL

THROUGHOUT THIS PROJECT, WE CREATED A SMALL, FAST-ENCRYPTING CHIP USING "HUMMINGBIRD" AS ITS CIPHER. THE CHIP WAS CREATED USING VHDL LANGUAGE AND HAD MANY MODULES THAT WORKED TOGETHER. THE DATA PATH HAD 16 BINARY DIGITS (OR BITS) AND INCLUDED UNCOMPLICATED SUBSTITUTIONS FOR CHANGING LETTERS (CALLED 'S-BOXES'), MIXING BITS ACROSS DIFFERENT LOCATIONS IN THE OUTPUT (CALLED 'LINEAR BIT MIXING'), AND MAKING SIMPLE CALCULATIONS WITH GROUPS OF TWO BITS (CALLED 'XOR' OPERATIONS). THE LFSRS USED TO GENERATE RANDOM NUMBERS WERE EACH BASED ON 16 BITS, WHILE THE BANKS USED TO KEEP THE STATE OF THE REGISTERS HAD FOUR REGISTERS, ENABLING THE ABILITY TO CYCLE THROUGH THEM SEQUENTIALLY. THE KEY SCHEDULER TOOK THE KEY (256 TOTAL BITS) AND SPLIT IT UP INTO SMALLER GROUPS (SUBKEYS) THAT WERE THEN SENT TO DIFFERENT ENCRYPTING MODULES BY THE CONTROLCORE FINITE STATE MACHINE. FINALLY, AFTER CONNECTING ALL THESE PARTS TOGETHER, WE BUILT THE ENTIRE CHIP IN ONE FILE AND CHECKED IT AGAINST OUR TESTING PROCEDURES, VERIFYING THAT ALL THREE PHASES WORKED AS EXPECTED AND THAT THE MESSAGE COULD BE DECRYPTED CORRECTLY. THIS MEANS WE SUCCESSFULLY PRODUCED A VERY EFFICIENT, USER-FRIENDLY WAY TO IMPLEMENT HARDWARE-BASED CRYPTOGRAPHY ON FPGAS, WITH MODULARISATION AND SEQUENCING AS THE MAIN DESIGN PRINCIPLES.

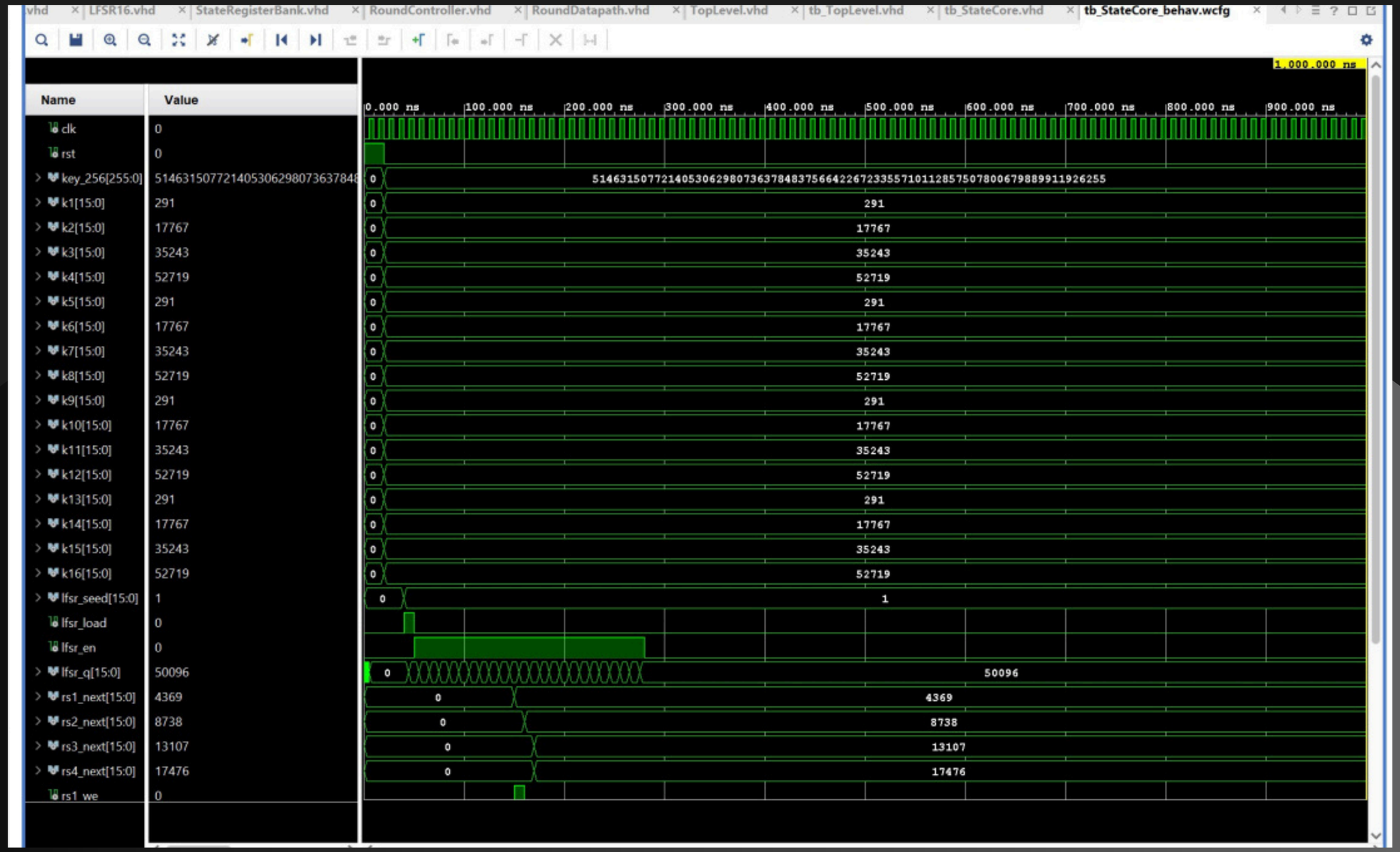
TESTING

INITIALLY FOR THE SIMULATION, WE PREDICTED THAT WHEN RESET, THE REGISTERS, COUNTERS, AND OUTPUTS WILL HAVE THEIR STATES RETURN BACK TO QUIESCENT STATE WITH NO ENABLE SIGNALS ASSERTED. AFTER THE RESET IS DE-ASSERTED AND THE START SIGNAL IS ASSERTED, WE EXPECT THE SYSTEM TO ENTER INTO AN INITIALIZATION PHASE WHERE THE LFSR SEED WILL BE LOADED, REGISTER WRITE-ENABLES WILL BE ACTIVATED, AND THE LFSR WILL BE ADVANCED UNTIL THE INITIALIZATION COUNT IS REACHED. AFTER INITIALIZATION, WE EXPECT THE CONTROLLER TO PROGRESS THROUGH THE FOUR ENCRYPTION STAGES (ENC_R1 TO ENC_R4), EACH TIME ASSERTING ONLY ONE STATE REGISTER'S WRITE-ENABLE SIGNAL, WHILE INCREMENTING THE ROUND COUNT. AFTER ALL ROUNDS ARE COMPLETED, WE EXPECT THE CONTROLLER TO PROCEED TO THE DONE STATE AND PULSE THE READY SIGNAL, DE-ASSERT THE BUSY SIGNAL, AND RETURN TO THE IDLE STATE. THE WAVEFORM SHOULD SHOW A CLEAR PROGRESSION THROUGH THE VARIOUS STATES BY PROPERLY TRANSITIONING THE SIGNALS AS PER THE CONTROLLER'S REQUIREMENTS.

RESULT

THE GENERATED WAVEFORM CLOSELY REFLECTS THE HYPOTHESIZED BEHAVIOR AND CONFIRMS THAT THE CONTROLCORE FSM AND THE STRUCTURAL MODULES ARE FUNCTIONING CORRECTLY. AT THE BEGINNING OF SIMULATION, THE SYSTEM BEHAVES EXACTLY AS EXPECTED: THE RST SIGNAL IS HIGH, AND ALL INTERNAL OUTPUTS REMAIN STABLE AND INACTIVE. DURING THIS RESET INTERVAL, THE SYSTEM MAINTAINS A PREDICTABLE STATE WHERE NO ENABLE OR WRITE SIGNALS ARE ASSERTED. THIS CONFIRMS THAT BOTH THE LFSR AND THE REGISTER BANK ARE PROPERLY SYNCHRONIZED WITH THE RESET MECHANISM OF THE DESIGN.

RESULT


vhd
LFSR16.vhd
StateRegisterBank.vhd
RoundController.vhd
RoundDatapath.vhd
TopLevel.vhd
tb_TopLevel.vhd
tb_StateCore.vhd
tb_StateCore_behav.wcfg
Name
Value
clk 0
rst 0
key_256[255:0] 51463150772140530629807363784837566422672335571011285750780067988991192625
k1[15:0] 291
k2[15:0] 17767
k3[15:0] 35243
k4[15:0] 52719
k5[15:0] 291
k6[15:0] 17767
k7[15:0] 35243
k8[15:0] 52719
k9[15:0] 291
k10[15:0] 17767
k11[15:0] 35243
k12[15:0] 52719
k13[15:0] 291
k14[15:0] 17767
k15[15:0] 35243
k16[15:0] 52719
lfsr_seed[15:0] 1
lfsr_load 0
lfsr_en 0
lfsr_q[15:0] 50096
rs1_next[15:0] 4369
rs2_next[15:0] 8738
rs3_next[15:0] 13107
rs4_next[15:0] 17476
rs1_we 0
1,000.000 ns
0.000 ns
100.000 ns
200.000 ns
300.000 ns
400.000 ns
500.000 ns
600.000 ns
700.000 ns
800.000 ns
900.000 ns

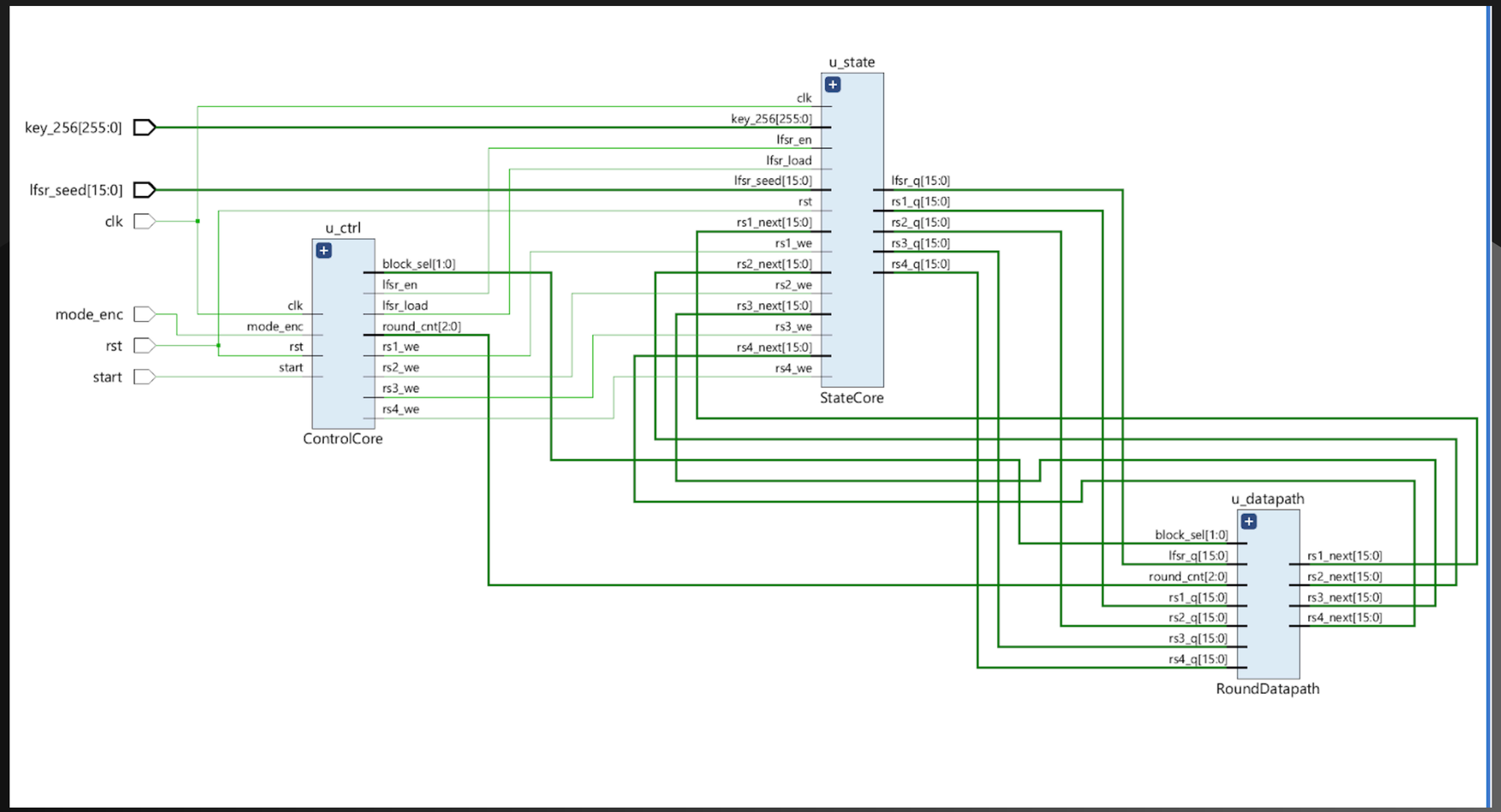
ANALYSIS

AFTER THE RESET IS ISSUED, THE SYSTEM INITIALLY ENTERS IDLE STATE UNTIL START SIGNAL BECOMES ACTIVE CAUSING FSM TO PERFORM INITIALISATION SEQUENCE. THE WAVEFORM DEMONSTRATES THE EXPECTED INITIALISATION BEHAVIOUR; DURING THIS FIRST CYCLE, LFSR LOAD SIGNAL GOES TO ACTIVE STATE, ALLOWING ALL REGISTER WRITE ENABLES TO GO TO ACTIVE STATE, SO THE SEED GETS LOADED INTO R0 AND INITIALISING THE STATE REGISTERS. THE RESULTING CYCLES CONTINUE FOR FURTHER INITIALISATION WHERE THE LFSR AND STATE REGISTERS GET UPDATED. AFTER COMPLETING THE FINALISED INITIALISATION THE SYSTEM TRANSITS INTO THE ENCRYPT STAGES (ENC_R1 - ENC_R4) OF OPERATION, WHERE ONLY THE CORRECT REGISTER WRITE ENABLE WILL BE IN AN ACTIVE STATE, WITH THE BLOCK SELECTOR TRANSITIONING AUTOMATICALLY. DURING THE ENCRYPT STAGES THE LFSR CONTINUES GENERATING RANDOM VALUES VERIFYING THE EXPECTED OPERATION. IN THE LAST STAGES, THE SYSTEM ACTIVATES DONE STATE, PRODUCING A READY SIGNAL FOR ONE CYCLE, WHILE BUSY REMAINS LOW, WHICH ALLOWS FOR A RETURN TO IDLE STATE. THIS INDICATES SUCCESSFUL COMPLETION OF ENCRYPTION AND CONFIRMS THAT SIMULATION WAS PERFORMED ACCURATELY.

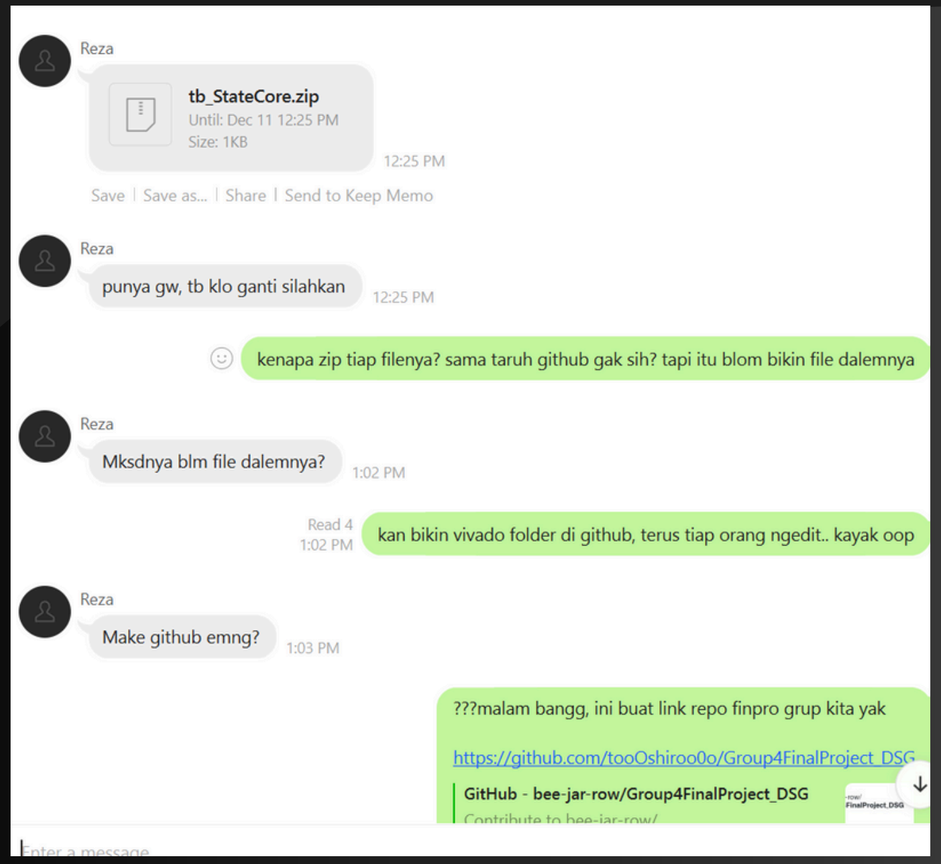
CONCLUSION

THE DESIGN AND IMPLEMENTATION OF THE HUMMINGBIRD LIGHTWEIGHT CRYPTO CORE IN VHDL GAVE US INSIGHT INTO HOW TO DEVELOP, CONTROL, AND VERIFY THE OPERATION OF HARDWARE-BASED CRYPTOGRAPHIC MODULES. BY BUILDING THE KEY COMPONENTS (KEY SCHEDULER, LFSR, STATE REGISTERS, AND MULTIPLE-STAGE CONTROLLER), WE WERE ABLE TO LEARN HOW THESE DIGITAL MODULES WORK TOGETHER TO CREATE AN ENCRYPTION DATA PATH. THE SIMULATION RESULTS CONFIRMED THAT OUR PROPOSED DESIGN FUNCTIONED AS INTENDED, WITH SIGNALS SUCH AS LFSR_LOAD, LFSR_EN, AND BLOCK_SEL DISPLAYING THE CORRECT TIMING AND COORDINATION. OUR DESIGN WAS A LESS COMPLEX IMPLEMENTATION OF THE UNDERLYING PRINCIPLES OF THE HUMMINGBIRD CIPHER, WHICH IS DESIGNED FOR ENERGY-EFFICIENT USE IN RESOURCE-LIMITED DEVICES (E.G., RFID TAGS, IOT SENSORS AND SIMILAR DEVICES). THROUGH THIS PROJECT, WE GAINED MORE OF AN UNDERSTANDING OF HARDWARE-LEVEL CRYPTOGRAPHY AND HOW TO DESIGN A SYSTEM IN VHDL, WHILE ALSO GAINING EXPERIENCE IN THE PROCESSES OF IMPLEMENTATION AND VERIFICATION OF A FUNCTIONING CRYPTO-CORE.

APPENDIX A : PROJECT SCHEMATIC



APPENDIX B : DOCUMENTATION



REFERENCES

- [1] ENGELS, DANIEL & FAN, XINXIN & GONG, GUANG & HU, HONGGANG & SMITH, ERIC. (2010). HUMMINGBIRD: ULTRA-LIGHTWEIGHT CRYPTOGRAPHY FOR RESOURCE-CONSTRAINED DEVICES. FINANC. CRYPTOGR. DATA SECUR.. 6054. 3-18. 10.1007/978-3-642-14992-4_2.
- [2] S. MAMMOU, D. BALOBAS AND N. KONOFAOS, "A VHDL IMPLEMENTATION OF THE HUMMINGBIRD CRYPTOGRAPHIC ALGORITHM," 2017 PANHELLENIC CONFERENCE ON ELECTRONICS AND TELECOMMUNICATIONS (PACET), XANTHI, GREECE, 2017, PP. 1-4, DOI: 10.1109/PACET.2017.8259979. KEYWORDS: {ENCRYPTION;REGISTERS;CIPHERS;ALGORITHM DESIGN AND ANALYSIS;TRANSFORMS;CLOCKS;VHDL;HUMMINGBIRD;CRYPTOGRAPHIC ALGORITHM},
- M. -Q. XIAO, X. SHEN, Y. -Q. YANG AND J. -Y. WANG, "LOW POWER IMPLEMENTATION OF HUMMINGBIRD CRYPTOGRAPHIC ALGORITHM FOR RFID TAG," 2010 10TH IEEE INTERNATIONAL CONFERENCE ON SOLID-STATE AND INTEGRATED CIRCUIT TECHNOLOGY, SHANGHAI, CHINA, 2010, PP. 581-583, DOI: 10.1109/ICSICT.2010.5667322. KEYWORDS: {CLOCKS;ALGORITHM DESIGN AND ANALYSIS;ENCRYPTION;RADIOFREQUENCY IDENTIFICATION;POWER DEMAND;OPTIMIZATION},
- D. ENGELS, M. J. O. SAARINEN, P. SCHWEITZER, AND M. E. SMITH, "THE HUMMINGBIRD-2 LIGHTWEIGHT AUTHENTICATED ENCRYPTION ALGORITHM," IN PROCEEDINGS OF THE 7TH INTERNATIONAL CONFERENCE ON RFID SECURITY AND PRIVACY (RFIDSEC), USA, 2011.
- F. M. NASCIMENTO, F. M. DOS SANTOS, AND E. D. MORENO, "A VHDL IMPLEMENTATION OF THE LIGHTWEIGHT CRYPTOGRAPHIC ALGORITHM HIGHT," IN PROCEEDINGS OF THE 9TH FORUM ON SPECIFICATION AND DESIGN LANGUAGES (FDL), 2015, PP. 1–8.

THANKYOU