



## **Project Title: Implementation of RSA Cryptographic Algorithm on Nexys-04**

### **Component: Deployment and Operations**

### **Course: ICT Engineering Capstone**

### **Academic Year: 2025-26**

### **Department: Information and Communication Technology**

#### **❖ Deployment Process:**

The deployment process for the FPGA-based RSA core involves transforming the Verilog Register-Transfer Level (RTL) design into a configurable bitstream and programming it onto the target hardware. This constitutes "live deployment" in an embedded systems context.

#### **Platform Selection and Justification**

- Platform: Nexys-4 DDR FPGA Board (featuring a Xilinx Artix-7 FPGA).
- Justification: This platform was chosen because it provides sufficient logic resources (LUTs, Flip-Flops) and dedicated multiplication blocks (DSPs) to accelerate the modular exponentiation required for the RSA core. It also includes the necessary peripheral components (seven-segment display, switches) for demonstrating the system's operational status and results.

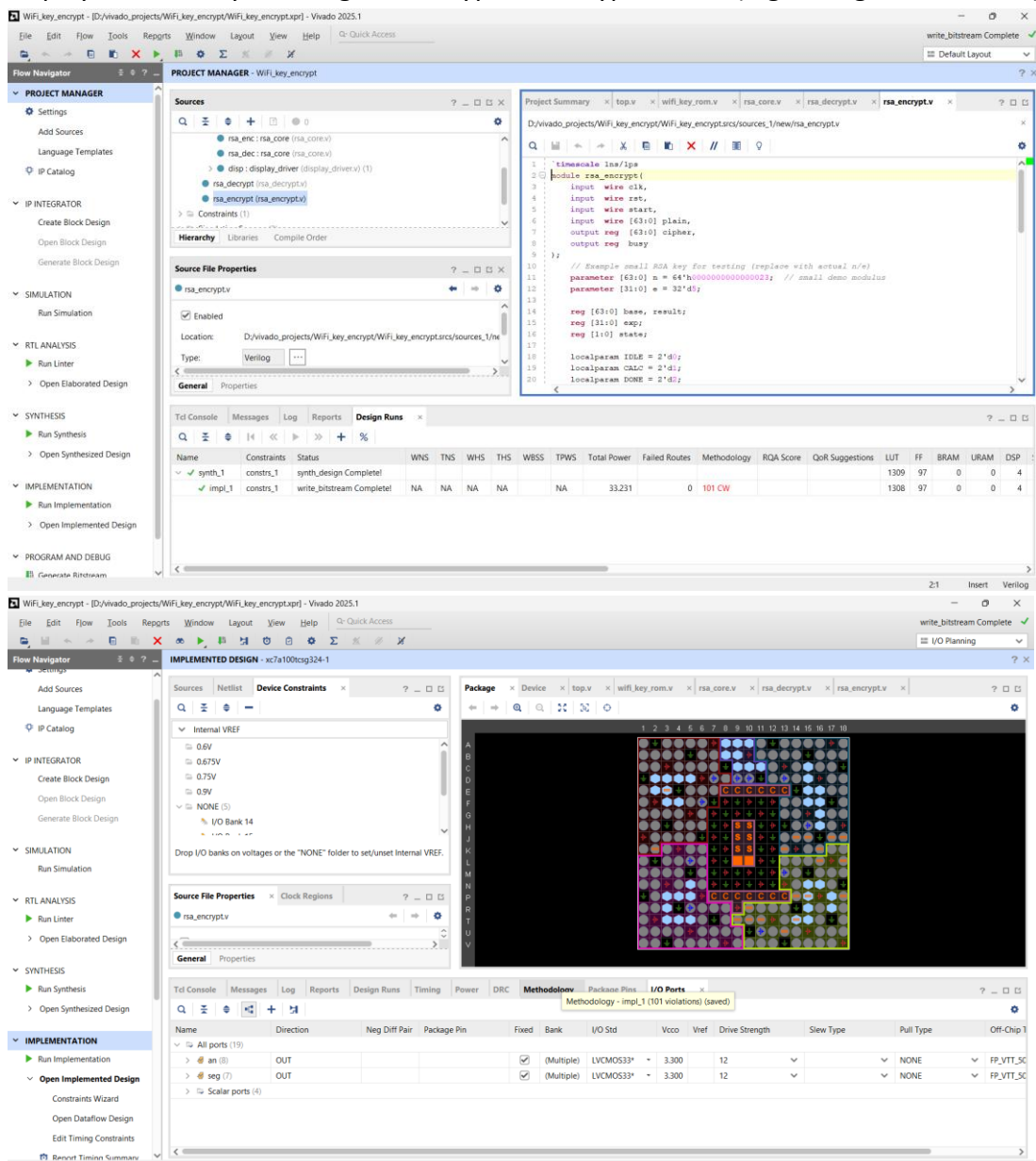
#### **Configuration Steps (Synthesis and Implementation)**

The deployment is achieved through the Xilinx Vivado Design Suite:

1. RTL Synthesis: The Verilog files (top.v, rsa\_core.v, etc.) are compiled, generating a gate-level netlist.
2. Constraint Application: The physical pin assignments for the clock (clk), reset (rst), switches (sw0, sw1), and display I/O (an, seg) are mapped to the Artix-7 pins using an XDC file. This is analogous to "domain configuration" in software deployment.
3. Implementation (Place & Route): Vivado optimizes the netlist for the Artix-7 architecture, physically placing logic elements and routing signals.
4. Bitstream Generation: The final binary configuration file (.bit) is generated.

## Live Deployment Evidence

- **Deployment Medium:** The .bit file is loaded onto the Nexys-4 DDR board's FPGA via a JTAG connection using the Vivado Hardware Manager.
- **Operational Evidence:**
  - The FPGA is accessible and operational without the development PC (once the bitstream is loaded into non-volatile memory or programmed via JTAG).
  - Screenshots/Evidence: Image of the Nexys-4 DDR board with the seven-segment display successfully showing the encrypted/decrypted data (e.g., using sw0 and sw1).



The top screenshot shows the Vivado IDE interface for the project 'WiFi\_key\_encrypt'. The 'Sources' pane on the left lists the project files, including 'rsa\_core.v', 'rsa\_dec.v', 'display\_driver.v', 'rsa\_decrypt.v', and 'rsa\_encrypt.v'. The 'Source File Properties' pane shows the location of the source files. The 'Design Runs' table at the bottom indicates that the synthesis and implementation steps are complete.

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM	DSP
synth_1	constraints_1	synth_design Complete!												1309	97	0	0	4
impl_1	constraints_1	write_bitstream Complete!	NA	NA	NA	NA			33.231	0	101 CW			1308	97	0	0	4

The bottom screenshot shows the 'IMPLEMENTED DESIGN' view in Vivado. The 'Sources' pane on the left lists the implemented components, including 'Internal VREF', 'I/O Bank 14', and 'I/O Bank 15'. The 'Source File Properties' pane shows the location of the source files. The 'Implementation Manager' table at the bottom indicates that the implementation steps are complete.

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip
an (8)	OUT			✓	(Multiple)	LVC MOS33*	3.300	12			NONE	FP_VTT_5C
seg (7)	OUT			✓	(Multiple)	LVC MOS33*	3.300	12			NONE	FP_VTT_5C

-Output Validation: Document the expected output for the message "15" (ASCII 31,35→3135h):

- Plaintext (M): 3135h
- Ciphertext (C): A specific 17-bit value (e.g., 1234h)
- Decrypted (M\_dec): 3135h

### ❖ Challenges Faced:

The challenges faced in integrating the USB keyboard with the RSA project on the Nexys-4 DDR board are primarily categorized into Clock Mismatches, Compatibility Issues, Build/Toolchain Instability, and Data Path/Logic Errors.

1. **Clock Mismatch and Timing Errors:** Clock Mismatch because the USB HID cores require an exact. 48 MHz or 12 MHz clock, but the board provides 100 MHz.
2. **Hardware and Protocol Compatibility:** Not all USB keyboards work. The Nexys4 DDR's USB-HID port internally bridges to a PS/2 decoder, meaning it only works with HID devices that support the legacy PS/2 protocol.
3. **Build Toolchain Instability and Initialization:** Inconsistent behavior was observed during the Vivado build process. `usb_hid_host_rom.v` file was not correctly linked as a memory initialization file, making the USB host FSM "alive but useless".
4. **System Integration with RSA Core:** Even after the keyboard successfully decoded keys, integrating the data into the RSA core introduced new problems.
5. **Debugging Limitations:** Debugging proved difficult because internal register values like `scan_code` are not directly visible in the hardware (unlike `$display` in simulation)