

# GH09.B.3.crypto – Encryption

---

## General Description

---

This is an Encryption algorithm based on Advanced Encryption Standard (defined in the Federal Information Processing Standards Publication 197). AES is a block cipher algorithm used and defined by the U.S. government.

## Features

---

- AES-128 cypher
  - 128 bit cryptographic key.
  - 128 bit data blocks.

## Block Diagram

---

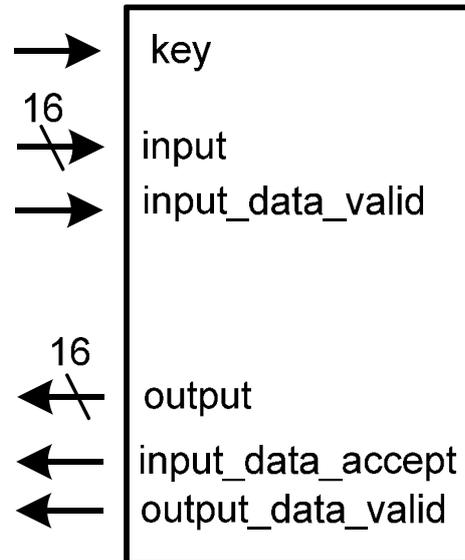


Figure 1 – Block diagram of the AES encryption core and the input and output signals

## Details

---

The AES-128 design is based on the (FIPS 197 specification) for 128 bit key and blocks of data. The full description of the algorithm is in FIPS 197. We will simply highlight which sections are of key importance when implementing this design. The references will be in the form:

FIP197-X.Y pg(s) A-B

Where FIP197 refers to the document, X is the section number, Y is the subsection number, A is the start page for the page range, and B is the end page for the page range.

## Algorithm

---

For this design only the cipher portion of the algorithm must be built. This portion of the algorithm converts plaintext (FIP197-2.1 pg8) to cyphertext (FIP197-2.1 pg 8).

The input to the chip will be in the form as described in FIP197-3.2 and FIP197-3.3 pgs 8 and 9. Since both input and output bits to the chip are in parallel, assume that the least significant bit of a port corresponds to "Input bit sequence" 0 as in Figure 2 FIP197-3.3 pg 9 also known as input0. Respectively, the most significant bit of a port corresponds to "Input bit sequence" 127 also known as input127.

The AES cipher algorithm is defined on pages FIP197-5 pgs 13-20.

## Chip Description

---

### Basic operation

The algorithm, as specified above, and the input and output protocols define most of the design with the exception of the basic operation of the device. To use the aes encryption the following steps are taken:

- Reset the chip
- Set the *key pin* high to indicate that the next 8 input packets will make up the encryption key
- Send the encryption key through the input port as specified in the Input Protocol section
- Set the *key pin* low to indicate that data will now be sent
- Send in the data on the input port pins

The reason for this setup is to keep the number of pins needed for data input relatively low, and thus, allow smaller devices to implement the AES encryption.

## Input Protocol

Inputs are received using a simple handshaking protocol. The pins of importance are the input (parallel port), the input\_data\_ready, the input\_data\_valid, the reset, and the clock. Figure 2 shows the basic communication setup between receiver (this design) and sender (external environment), and Figure 3 shows a waveform for a simple transfer between sender and receiver assuming the sender has one 8 bit packet to send (note that the input data width in the examples is 8 bits as opposed to 128 bits in the actual AES design shown in Figure 1 above).

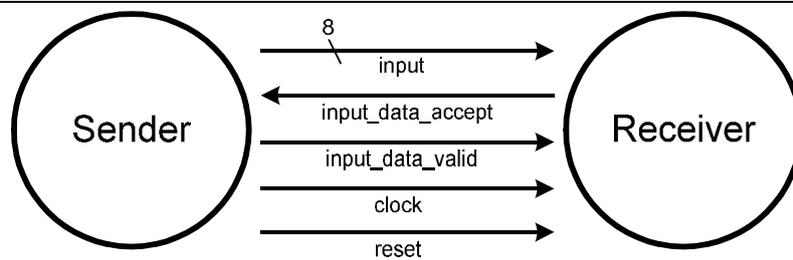


Figure 2 – The interface signals between the sender (the external environment) and the receiver (this design).

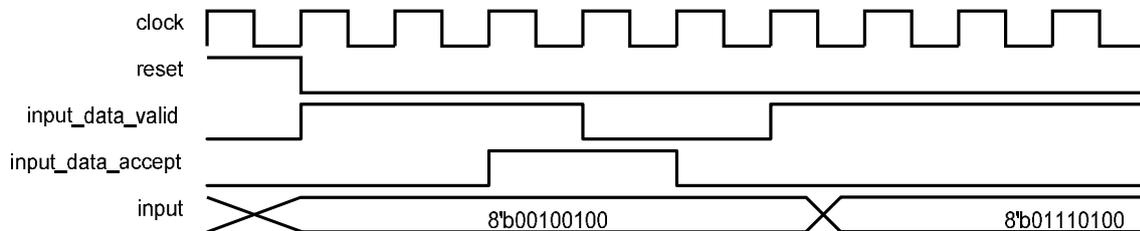


Figure 3 – a waveform for the communication between sender and receiver for one packet.

The assumption in this simple send receive protocol is that the shared clock between sender and receiver are synchronized. This simple interface removes much of the circuitry that would be needed to interface between devices.

Figure 4 and 5 show the finite state diagrams of both the sender and receiver.

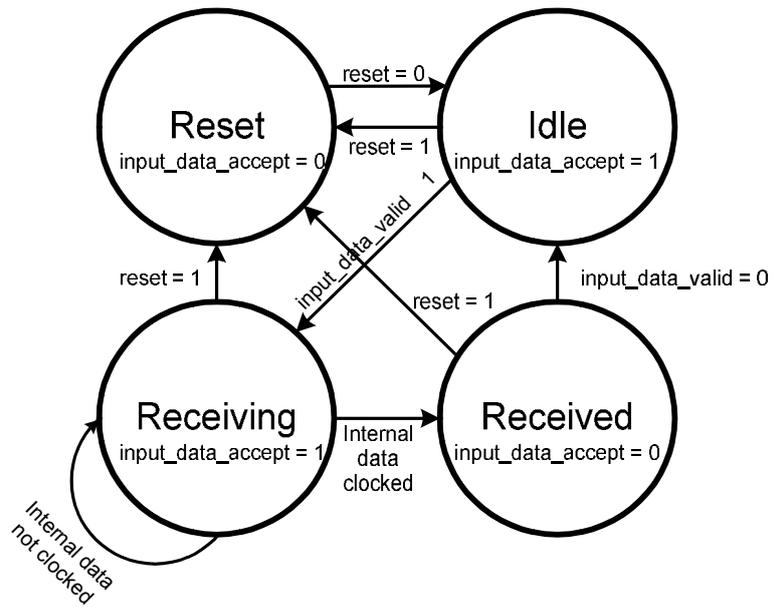


Figure 4 – Finite state machine for the receiver.

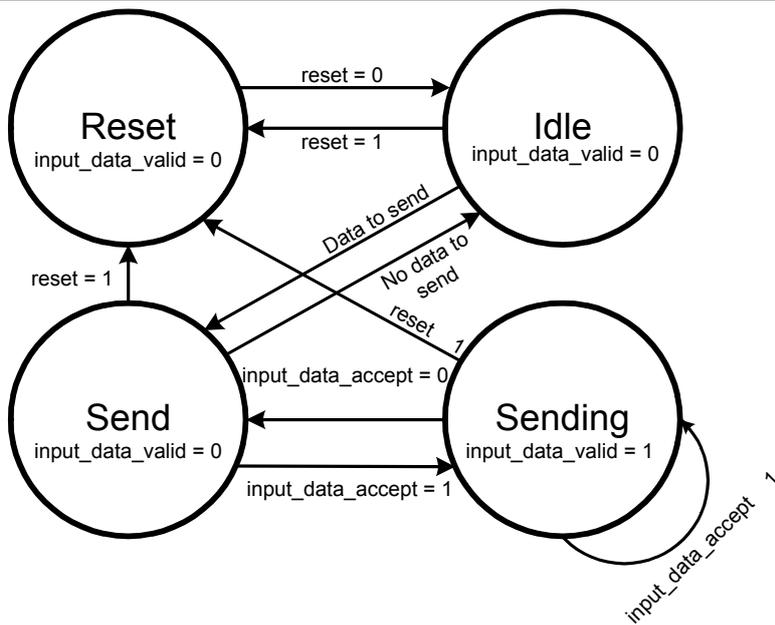
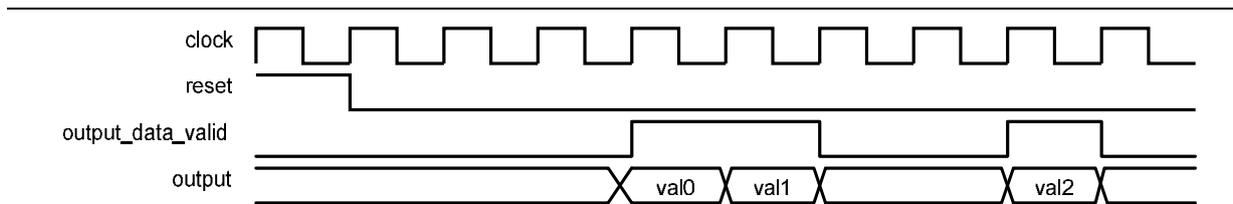


Figure 5 – Finite state machine for the sender.

## Output Protocol

The output protocol is a very simple protocol. When the output\_data\_valid signal is high for one clock cycle, then the output bus contains a value. Figure 6 shows a waveform where three output values (val0, val1, and val2) are sent out of from the design.



*Figure 6 – A waveform showing the output being sent and the corresponding output\_data\_valid signal.*

---