
# I8255 PPI Project

---

Submitted by:

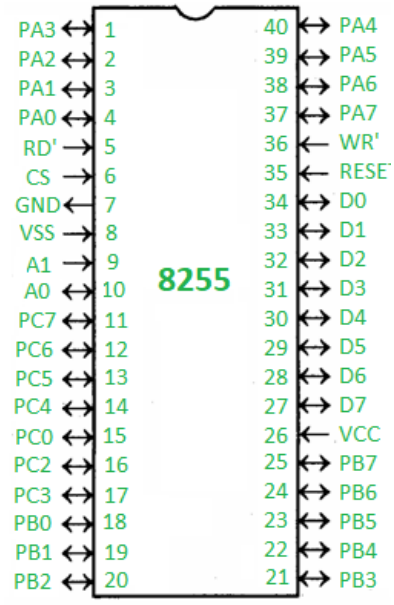
Nada Ihab Ahmed Mohamed Abdelgawad

Section: 3

2081/2019

Cairo, Egypt

# Introduction:




The **Intel 8255** (or **i8255**) Programmable peripheral interface (PPI) chip was developed and manufactured by Intel in the first half of the 1970s for the 8080 microprocessor. The 8255 provides 24 parallel input/output lines with a variety of programmable operating modes.

The 8255 gives a CPU or digital system access to programmable parallel I/O The 8255 has 24 input/output pins. These are divided into three 8-bit ports (A, B, C). Port A and port B can be used as 8-bit input/output ports. Port C can be used as an 8-bit input/output port or as two 4-bit input/output ports or to produce handshake signals for ports A and B.

The three ports are further grouped as follows:

1. Group A consisting of port A and upper part of port C.
2. Group B consisting of port B and lower part of port C.

There are two basic operational modes of 8255:

- Bit Set / Reset mode (BSR mode).
- Input / Output mode (I/O mode).

## Verilog Code structure:

My Verilog code is divided to 6 modules divided as follows:

Module for port A, port B, port C, group A, group B, and the main module (PPI_8255).

# Simulation results:

Here are some test cases from the test bench and wave (WRITE):

- The following figure shows some test cases for writing data to port A then port B then port C.

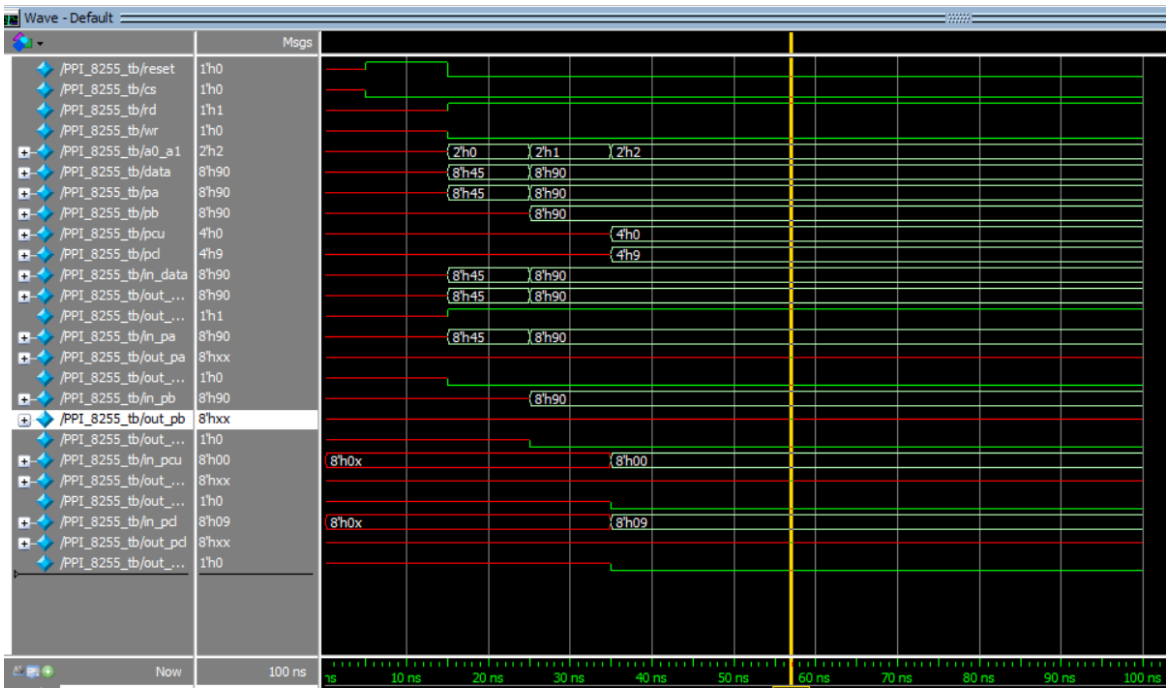
```
initial
begin
$monitor($time,,"%h %b %b %b %b %b %b %b %b %b",reset, cs, rd, wr, a0_a1, data, pa, pb, pcl, pcu);
#5
reset = 1;
cs = 0;
#10
reset = 0;
cs = 0;
rd = 1;
wr = 0;
a0_a1 = 2'b00;
out_valid_data = 1;
out_valid_pa = 0;
out_data = 8'h45;
#10
a0_a1 = 2'b01;
out_data = 8'h90;
out_valid_pb = 0;
#10
a0_a1 = 2'b10;
out_valid_pcu = 0;
out_valid_pcl = 0;

end
```

- The following figure shows simulation results from the transcript window:

```
VSIM 6> run
#                    0 x x x x xx xxxxxxxx xxxxxxxx xxxxxxxx xxxx xxxx
#                    5 1 0 x x xx xxxxxxxx xxxxxxxx xxxxxxxx xxxx xxxx
#                   15 0 0 1 0 00 01000101 01000101 xxxxxxxx xxxx xxxx
#                   25 0 0 1 0 01 10010000 10010000 10010000 xxxx xxxx
#                   35 0 0 1 0 10 10010000 10010000 10010000 1001 0000
```

- The following figure shows simulation result from the wave:



Here are some test cases from the test bench and wave (READ):

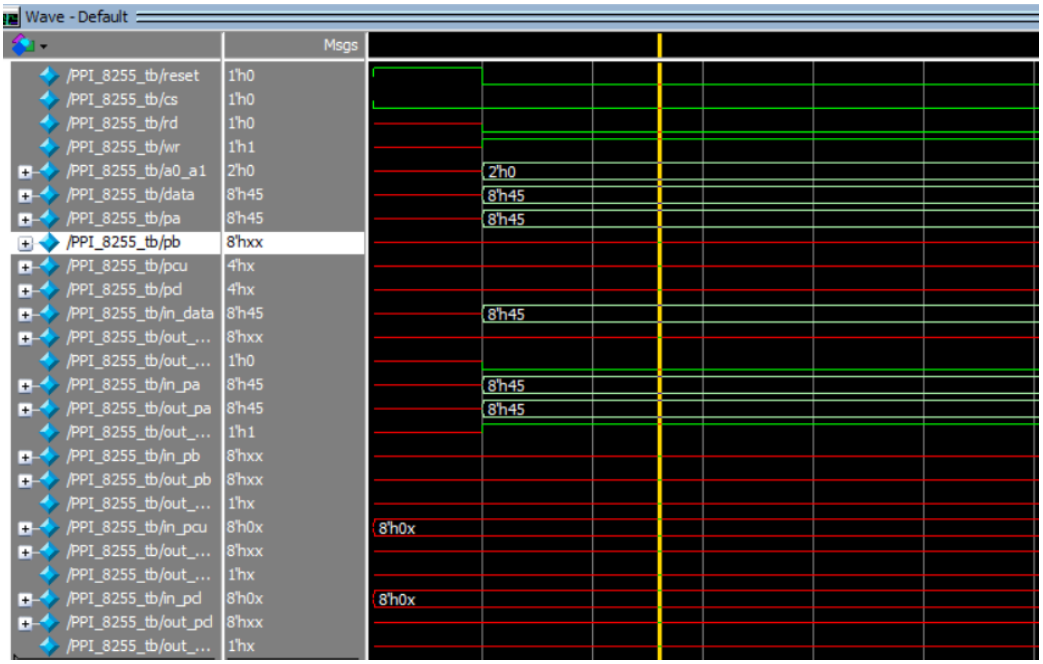
- The following figure shows some test cases for reading from port A then port B then port C.

```
initial
begin
$monitor($time,,"%b %b %b %b %b %b %b %b %b %b",reset, cs, rd, wr, a0_a1, data, pa, pb, pcu, pcl);
reset = 1;
cs = 0;
#10
reset = 0;
rd = 0;
wr = 1;
a0_a1 = 2'b00; //read from port A
out_valid_data = 0;
out_valid_pa = 1;
out_pa = 8'h45;
```

- The following figure shows simulation results from the transcript window:

```
VSIM 57> run
#                    0 1 0 x x xx xxxxxxxx xxxxxxxx xxxxxxxx xxxx xxxx
#                   10 0 0 0 1 00 01000101 01000101 xxxxxxxx xxxx xxxx
```

- The following figure shows simulation result from the wave:



Here are some test cases from the test bench and wave (BSR MODE):

- The following figure shows some test cases for setting and clearing some bits in portc

```
initial
begin
$monitor($time,,"%b %b %b %b %b %b %b %b %b %b",reset, cs, rd, wr, a0_a1, data, pa, pb, pcu, pcl);
#5
reset = 1;
cs = 0;
#10
reset = 0;
cs = 0;
rd = 1;
wr = 0;
a0_a1 = 2'b11;
out_valid_data = 1;
out_data = 8'h01; // set bit number zero in portc (BSR mode)
out_valid_pcu = 0;
out_valid_pcl = 0;
#5
out_data = 8'h0f; // set bit number seven in portc
#5
out_data = 8'h02; // clear bit number one in portc

end
```

- The following figure shows simulation results from the transcript window:

```
VSIM 25> run
#                    0 x x x x xx xxxxxxxx xxxxxxxx xxxxxxxx xxxx xxxx
#                    5 1 0 x x xx xxxxxxxx xxxxxxxx xxxxxxxx xxxx xxxx
#                   15 0 0 1 0 11 00000001 xxxxxxxx xxxxxxxx 0000 0001
#                   20 0 0 1 0 11 00001111 xxxxxxxx xxxxxxxx 1000 0000
#                   25 0 0 1 0 11 00000010 xxxxxxxx xxxxxxxx 0000 0000
```

- The following figure shows simulation result from the wave: