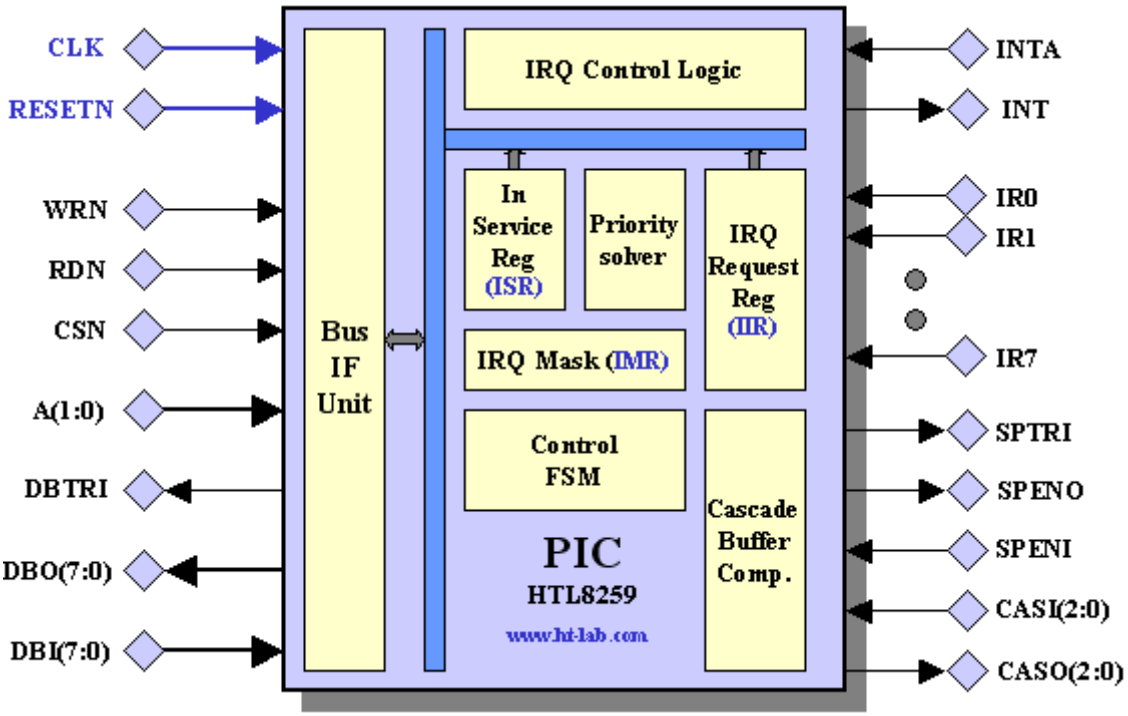
HTL8259

Programmable Interrupt Controller



Version 1.0a

1. Introduction

The HTL8259 is a synchronous implementation of the industry standard 8259/8259A Programmable Interrupt Controller (PIC). The PIC can be used to extend the number of interrupt lines on a microcontroller/processor.

The HTL8259 provide a wide range of vectored interrupt handling and ternination modes. The controller can be programmed for fixed, priority and polled mode of operation. Interrupt inputs can either be level or edge triggered. A single HTL8259 controller can handle up to 8 vectored priority interrupts. Without any glue logic multiple HTL8259 can be cascaded to provide upto 64 vectored priority interrupts. This number can be further increased by using the Poll command.

There are no software differences between the HTL8259 and a VLSI 8259 device. For this reason no programming information is provided in this document. Any commercial 8259/8259a datasheet can be used to obtain this information.

2. Directory Structure

The HTL8259 IP core is delivered in a single zip file. The directory structure after unzipping is as follows:

Directory	Contents
HTL8259\bin	Any executable files such as utilities (might be empty)
HTL8259\doc	Documentation and Datasheets
HTL8259\rtl	Synthesizable IP Core
HTL8259\testbench	Testbench files
HTL8259\Modelsim	Example script for Modelsim Simulator
HTL8259\synthesis	Synthesis scripts

3. Filelist

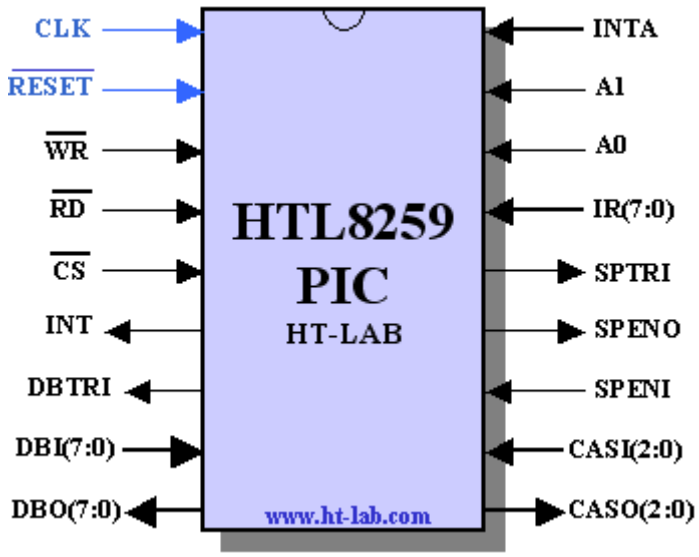
The HTL8259 RTL design is contained in 7 VHDL files.

Filename	Contains
pulselevel.vhd	Interrupt input logic
frontend_rtl.vhd	Interrupt Register, level/trigger selection
priority_rtl.vhd	Interrupt Priority logic
backend_rtl.vhd	Interrupt Service logic
wrctrl.vhd	Register logic and control
ctrl.vhd	Control FSM
htl8259a.vhd	Top Level

A further 3 files are used for the testbench.

Filename	Contains
htl8259_tester.vhd	Testbench Stimulus module
utils.vhd	Support utilities
htl8259_tb.vhd	Testbench, instantiates the HTL8259 and tester module

4. Pin Description



Symbol	Type	Function
CLK	I	Clock Input signal [Note1]
RESET	I	Active Low Reset signal
WR	I	Active Low Write Strobe
RD	I	Active Low Read Strobe
CS	I	Active Low Chip Select signal
A0/A1	I	2 bits Address Bus
DBI(7:0)	I	8-bits DataBus Input
DBO(7:0)	O	8-bits DataBus Output
DBTRI	O	DataBus Tri-State Signal, Active High drives the output bus
INT	O	Interrupt Output signal to processor
INTA	I	Interrupt Acknowledge signal from processor
IR(7:0)	I	Interrupt input signals
SPENI	I	SP/EN Input signal
SPENO	O	SP/EN Output signal
SPTRI	O	SP/EN Tri-State signal, Active High drives the output bus
CASI(2:0)	I	3-bits CAS input bus
CASO(2:0)	O	3-bits CAS output bus

Note1: All actions are on the Rising Edge of the system clock.

5. Simulation

The HTL8259 IP core is written in vendor neutral VHDL and as such can be simulated by any modern simulation tool.

The testbench instantiates the htl8259 together with a stimulus generation file.

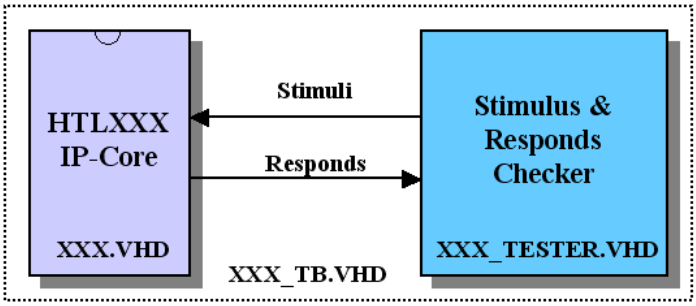


Figure: standard testbench setup

To run the simulation, execute the ***Modelsim\run.bat*** batch file from a CMD/Dosbox prompt. An equivalent Modelsim run.do file can be executed from within the Modelsim GUI.

Note that part of the validation was performed in software (8086 assembly language).

Part of the testbench output is shown below. See *expected_output.txt* in the Modelsim directory for the full listing.

```
#  run -all
# ------- Test Spurious Interrupt --------
# *** Request   IRm=00 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA :IRQ7 Vector 47
# ------- Test Non-Specific Interrupt --------
# *** Request  IRQ3 IRQ6  IRm=48 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA :IRQ3
Vector 43
# *** Request  IRQ3 IRQ6  IRm=48 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA :IRQ3
Vector 43
# --- Test Non-Specific Interrupt + Rotate ----
# *** Request  IRQ3 IRQ6  IRm=48 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA :IRQ3
Vector 43
# *** Request  IRQ3 IRQ6  IRm=48 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA :IRQ6
Vector 46
# *** Request  IRQ3 IRQ6  IRm=48 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA :IRQ3
Vector 43
# ------- Test Spurious Interrupt after rotate --------
# *** Request   IRm=00 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA :IRQ7 Vector 47
# --- Test Specific Interrupt ----
# *** Request  IRQ3 IRQ6  IRm=48 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA :IRQ3
Vector 43
# *** Request  IRQ3 IRQ6  IRm=48 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA :IRQ3
Vector 43
# --- Test Specific Interrupt + Rotate----
# *** Request  IRQ3 IRQ6  IRm=48 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA :IRQ3
Vector 43
# *** Request  IRQ3 IRQ6  IRm=48 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA :IRQ6
Vector 46
# *** Request  IRQ3 IRQ6  IRm=48 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA :IRQ3
Vector 43
# --- Test Automatic EOI Edge triggered ----
# *** Request  IRQ0 IRQ3 IRQ6 IRQ7  IRm=C9 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA
:IRQ0 Vector 40
# *** Request  IRQ0 IRQ3 IRQ6 IRQ7  IRm=C8 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA
:IRQ3 Vector 43
# *** Request  IRQ0 IRQ3 IRQ6 IRQ7  IRm=C0 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA
:IRQ6 Vector 46
# *** Request  IRQ0 IRQ3 IRQ6 IRQ7  IRm=80 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA
:IRQ7 Vector 47
# --- Test Automatic EOI + Rotate ----
# *** Request  IRQ0 IRQ3 IRQ6 IRQ7  IRm=C9 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA
:IRQ0 Vector 40
# *** Request  IRQ0 IRQ3 IRQ6 IRQ7  IRm=C9 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA
:IRQ3 Vector 43
# *** Request  IRQ0 IRQ3 IRQ6 IRQ7  IRm=C9 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA
:IRQ6 Vector 46
# *** Request  IRQ0 IRQ3 IRQ6 IRQ7  IRm=C9 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA
:IRQ7 Vector 47
# --- Test Automatic EOI level Stuck IRQ0 ----
# *** Request  IRQ0 IRQ3 IRQ6 IRQ7  IRm=C9 ISm=00 MKm=00  IRs=00 ISs=00 MKs=00  -> 8086 INTA
```

7. Synthesis

The HTL8259 can be synthesized using any modern synthesis tool. The area and delay figures will depend on the used synthesis tool and constraints.

An example DOS synthesis batch file is provided for Xilinx XST. To execute the batch file navigate to the ***Synthesis*** directory and execute the ***runxst.bat*** batch file. Make sure that the **xst.exe** executable is in the search path. Part of the log file (HTL8259a.syr) based on ISE11.1 is shown below.

```
Device utilization summary:
---------------------------

Selected Device : 3s500ecp132-4


 Number of Slices:                      261  out of   4656     5%
 Number of Slice Flip Flops:            127  out of   9312     1%
 Number of 4 input LUTs:                481  out of   9312     5%
 Number of IOs:                          42
 Number of bonded IOBs:                  42  out of     92    45%
    IOB Flip Flops:                       8
 Number of GCLKs:                        1  out of     24     4%


Timing Summary:
---------------
Speed Grade: -4

   Minimum period: 15.271ns (Maximum Frequency: 65.484MHz)
   Minimum input arrival time before clock: 14.701ns
   Maximum output required time after clock: 20.025ns
   Maximum combinational path delay: 16.305ns
```

8. History

Version	Date	Changes
1.0	27/07/02	First Version
1.0a	22/11/09	Tools Update Check

9. Support

For any support issues please use support@ht-lab.com

Trademarks

Spartan and Xilinx are the registered trademarks of Xilinx Inc, San Jose, California, US.
Modelsim is a trademark of Mentor Graphics.