iCE40LM Philips IR Rx Solution

Reference Design

FPGA-RD-02159-1.2

February 2020

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice’s product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Tables

1. General Description

The iCE40LM Philips IR Rx reference design is a low power infrared receiver solution for mobile devices. It is designed to receive data from an IR receiver module, validate the received data, and send the validated data to the application processor. This reference design acts as data control, and buffering protocol conversion between the IR receiver module and the application processor. When the IR Rx Solution is used in a design, it allows the application processor to sleep for longer periods of time by reducing unnecessary communication between the sensors and the processor, thus saving power consumption by allowing the processor not to be in an "always on" state. The iCE40LM Philips IR Rx reference design is a configurable solution, available as either standalone off the shelf or fully customizable solution, making it an IR Rx standard agnostic solution.

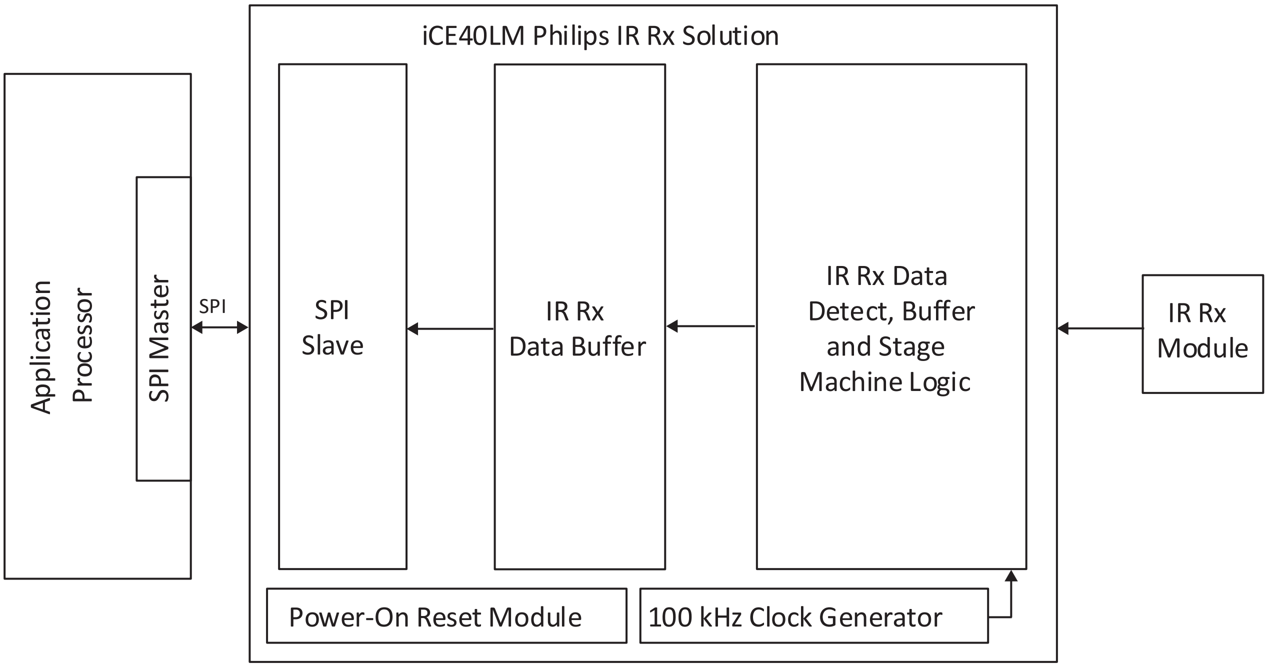


Figure 1.1. System Block Diagram

As a standalone solution, the iCE40LM Philips IR Rx Solution connects to the application processor's Serial Peripheral Interface Bus (SPI) with clock frequency set to 10.8 MHz. This enables a fast communication speed to/from the processor. The iCE40LM Philips IR Rx standalone solution prepares and sends the data as required by Philips IR Rx interface to the application processor. The data is recovered and converted to a bus format and then sent to the applications processor via a SPI bus.

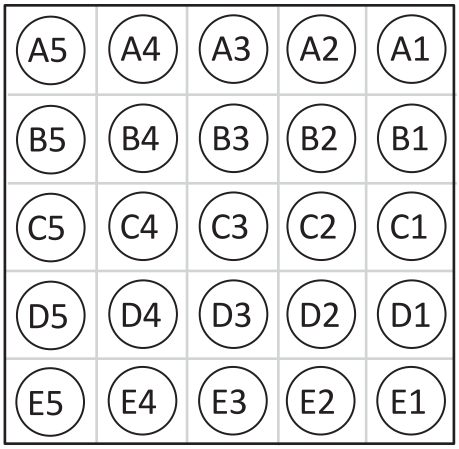
The iCE40LM Philips IR Rx standalone solution has a system operating frequency of 27 MHz, and a SPI bus frequency to application processor of 10.8 MHz. The SPI bus is configured to have a voltage of 1.8 V, and the IR Receiver Module is connected via a 3.3 V I/O.

The fully customizable solution capability of this solution is due to its FPGA based architecture. This capability is ideal, but not limited to users who would like to include additional IR Rx standards, change the data acquisition FIFO depth, create an I/O bridge between IR receiver modules and processor, or create additional custom logic.

The IR Rx reference design consumes only 418 LUTs. This allows it to fit in a device as small as iCE40LM1K.

Regardless of the solution type, the iCE40LM Philips IR Rx Solution has a core voltage of 1.2 V. It is available in a very small form-factor 25-pin WLCSP package. The package has 0.35 mm ball pitch, making the overall package size to be 1.71 mm x 1.71 mm that easily fit into a number of mobile devices such as smart phones. Other packages include .4 mm ball pitch with 36 balls (2.5x2.5 mm) or 49 balls (3x3 mm). The solution operates at industrial temperature range of -40 C to 100 C.

As a standalone solution, user simply obtains the solution which includes the device, Diamond Programmer software, and ready-for-download bitstream. As a fully customizable solution, user will obtain the device, the iCEcube2 design software, the programming software, and the source code of the IR Rx solution.



A5	A4	A3	A2	A1
B5	B4	B3	B2	B1
C5	C4	C3	C2	C1
D5	D4	D3	D2	D1
E5	E4	E3	E2	E1

Figure 1.2. Package Diagram (Balls Up)

2. Features

- Configurable IR Rx Solution
  - Configured to Philips IR Rx Interface
  - Default System frequency of 27 MHz
  - Power-On Reset capability
- Serial Peripheral Interface (SPI) Bus connection to Application Processor with the following Default settings:
  - Interface frequency of 10.8 MHz
  - Interface voltage of 1.8 V
  - Solution is a “slave” of the Application Processor
  - SPI slave mode CPOL = 1 and CPHA = 1 (mode “3”)
  - SPI slave features MSB first
- General
  - Core voltage of 1.2 V
  - I/O voltages of 1.8 V and 3.3 V
  - 25-pin WLCS at 1.69 mm x 1.69 mm with 0.35 mm pitch
  - Industrial (-40 C to 100 C) Grade

3. Applications

- IR Receiver Capable Devices
- Smart Phones
- Tablets
- Universal Remote Controls

4. Functional Block Diagram

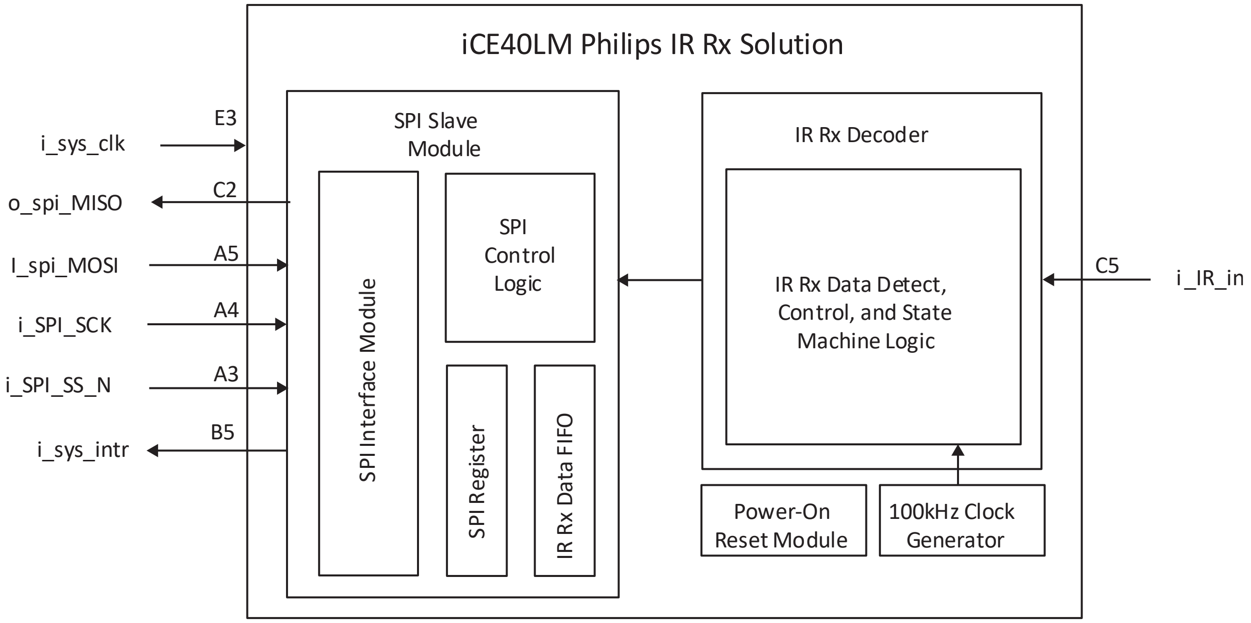


Figure 4.1. Functional Block Diagram

5. Specifications

5.1. Recommended Operating Conditions

Table 5.1. Recommended Operating Conditions

Symbol	Parameter	Min.	Typ.	Max.	Units
V_{CC}	Core Supply Voltage	See iCE40LM Family Data Sheet (FPGA-DS-02043).	1.2	See iCE40LM Family Data Sheet (FPGA-DS-02043).	V
$V_{CCIOVB1}$[1]	Bank 1 I/O Driver Supply Voltage	1.71	1.8	1.89	V
$V_{CCIOVB2}$[1]	Bank 2 I/O Driver Supply Voltage	3.14	3.3	3.46	V
t_{JUND}	Junction Temperature Operation	See iCE40LM Family Data Sheet (FPGA-DS-02043).	-	See iCE40LM Family Data Sheet (FPGA-DS-02043).	°C

Note:

1. Assumes operating under “off-the-shelf standalone1 solution”.

5.2. Power Supply Ramp Rates

See iCE40LM Family Data Sheet (FPGA-DS-02043).

5.3. Power-On-Reset Voltage Levels

See iCE40LM Family Data Sheet (FPGA-DS-02043).

5.4. ESD Performance

See iCE40LM Family Data Sheet (FPGA-DS-02043).

5.5. DC Electrical Characteristics

See iCE40LM Family Data Sheet (FPGA-DS-02043). Set the VCCIO values to the values stated in the Recommended Operating Conditions table.

5.6. Power Supply Current

See iCE40LM Family Data Sheet (FPGA-DS-02043).

5.7. Absolute Maximum Ratings

See iCE40LM Family Data Sheet (FPGA-DS-02043).

5.8. Performance Characteristics

Table 5.2. Performance Characteristics[1]

Symbol	Parameter	Min.	Typ.	Max.	Units
$F_{coremax}$	System Frequency			27	MHz
Tcoremaxdcd	Maximum duty cycle distortion for System Clock				%
F_{spimax}	SPI Bus Frequency			10.8	MHz
Tsuspi	SPI setup time				ns
Thdspi	SPI hold time				ns
Tcospi	SPI clock to out time				ns
Tcosysintr	o_sys_intr clock to out time				ns
Tsusn	i_ssn setup time				ns
Thdssn	i_ssn hold time				ns
Tsuirin	i_IR_in setup time				ns
Thdirin	i_IR_in hold time				ns
Tpor	Power-On Reset duration[2]		192		Cycles

Note:

1. Assumes operating under “off-the-shelf standalone1 solution”
2. Relative to System Frequency
3. All values are based on iCEcube2’s Timing Analyzer’s results. The design is not validated by test engineering.

5.9. FPGA Characteristics

See iCE40LM Family Data Sheet (FPGA-DS-02043). Note that once customization is performed, the values in Performance Characteristics may not be the same.

6. Pin Configuration and Function Descriptions

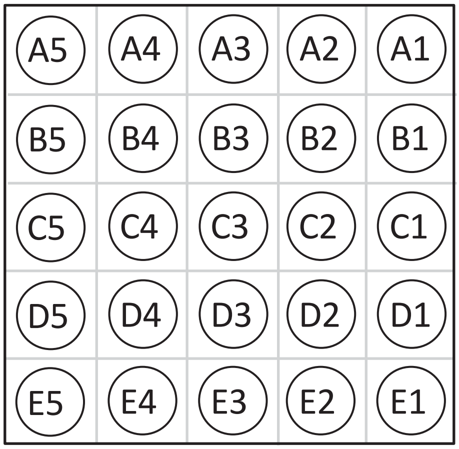


Figure 6.1. Bottom View of iCE40LM4K-SWG25TR

Table 6.1. Pin Function Description[1]

Pad Name	Port Name	Port Direction	Description
A1	General Purpose I/O	Input/Output	1.8 V I/O for user interface
A2	VCCIOVB1	Input	I/O Power Supply
A3	i_SPI_SS_N	Input	SPI bus slave select (Active Low)
A4	i_SPI_SCK	Input	SPI bus serial clock
A5	i_SPI_MOSI	Input	SPI bus serial data in to slave
B1	General Purpose I/O	Input/Output	1.8 V I/O for user interface
B2	GND	Input	Ground
B3	CRESET	Input	Configuration Reset (Active Low). See Datasheet
B4	VCC	Input	Core Power Supply
B5	o_sys_intr	Output	Interrupt to Application Processor (Active High)
C1	ice_SI	Output	Configuration Output to external SPI Memory
C2	o_SPI_MISO	Output	SPI bus serial data from slave
C3	CDONE	Output	Configuration Done. See Datasheet
C4	General Purpose I/O	Input/Output	3.3 V I/O for user interface
C5	i_IR_in	Input	Input from IR receiver module
D1	flsh_sclk	Input	Configuration Clock
D2	ice_SO	Input	Configuration Input from external SPI Memory
D3	General Purpose I/O	Input/Output	3.3 V I/O for user interface
D4	GND	Input	Ground
D5	General Purpose I/O	Input/Output	3.3 V I/O for user interface
E1	flsh_cs	Input	Configuration Chip Select (Active Low)
E2	VCCIOVB2	Input	I/O Power Supply
E3	i_sys_clk	Input	System Clock
E4	General Purpose I/O	Input/Output	3.3 V I/O for user interface
E5	General Purpose I/O	Input/Output	3.3 V I/O for user interface

Note:

1. Assumes operating under “off-the-shelf standalone1 solution”.

7. Theory of Operations

The iCE40LM Philips IR Rx Solution interfaces between an IR Receiver Module and an application processor. It receives Philips based IR Rx data through a 3.3 V I/O. The received data is then validated for Philips interfacing standard, formatted for the processor, and stored/buffered into a data FIFO. Once the data FIFO is no longer empty, an interrupt signal is sent to the processor to indicate that IR Rx data is present. The application processor is then expected to read the FIFO contents through the SPI bus. The whole process begins again when the next set of IR Rx data is present.

7.1. Functional Descriptions

This sub-section describes the function of each sub-block in inside the iCE40LM Philips IR Rx Solution. Many of these blocks have HDL module associated with them.

Philips IR Rx Top Level

The Philips IR Rx Top Level is found in SPI_RC6. This module contains the SPI Interface to Application Processor, and the IR Rx Decoder. It also contains a Power-On Reset (POR) module, and a 100 kHz clock generator to sample the IR Rx signal. The POR module initiates a system reset upon power up for Tpor number of cycles. The iCE40LM Philips IR Rx Solution operates after system reset has been completed.

SPI Interface to Application Processor

This module is used to interface between the iCE40LM Philips IR Rx Solution and the application processor. It is found in spi_reg module. It also contains the data FIFO for IR Rx data and logic that issues interrupt to the application processor to indicate that IR Rx data is present for the processor to read. The IR Rx data comes as a set of three bytes (mode, control, and info). As a result, the processor must read 3 bytes of data from the data FIFO for the complete data. In addition to data FIFO, this module contains a SPI register called VERSION.

IR Rx Decoder

This module is found in RC6_Decoder, and it receives an IR Rx data from IR receiver module via a 3.3 V I/O. The sampling clock used is the 100 kHz clock generated in fabric. This module contains state machine logic, edge detection circuit, and decoder logic to convert the serial data into three bytes of data (mode, control, and info). In addition, this module also indicates whether the received data are valid. All of these are then sent to the data FIFO in the SPI Interface to Application Processor module.

7.2. Block Descriptions

The purpose of this section is to provide detailed descriptions of each block of the iCE40LM Philips IR Rx Solution so as to assist users who want to use this solution as a building block for other IR Rx.

Top Level Module (SPI_RC6)

This module contains the SPI Interface to Application Processor, and the IR Rx Decoder. The HDL code begins with the POR logic, which is set to Tpor cycles, and followed by the 100 kHz clock generator. The two modules above are then instantiated and connected together.

The IR Rx Decoder is instantiated using RC6_Decoder module. This module samples IR Rx data from IR receiver module and detects/converts the serial data into Philips Rx format. The recovered data are then stored into the data FIFO in the SPI Interface to Application Processor module.

The SPI Interface to Application Processor module (found in spi_reg) provides communication to/from the processor. When the FIFO is not empty, this module issues an interrupt to the processor so that the processor can read the data. Once read command is issued, the data are sent to the processor via SPI interface.

IR RX Decoder (RC6_Decoder)

This module is found in the RC6_Decoder file. It is used to detect and recover serial data into Philips Rx format. The interface with the IR Receiver Module is a 3.3 V I/O, and the sampling clock used is 100 kHz clock generated in fabric.

Table 7.1 summarizes the ports to/from this module.

Table 7.1. Ports To/From IR Rx Decoder Module

Port Name	Direction	Description
o_ir_mode[2:0]	Output	IR Rx Info Mode Data from RC6 Decoder
o_ir_control_field[7:0]	Output	IR Rx Control Field Data from RC6 Decoder
o_ir_info_field[7:0]	Output	IR Rx Info Field Data from RC6 Decoder
o_ir_data_vld	Output	Determines whether the IR Rx data is valid (Active HIGH)
o_ir_data_error	Output	Determines whether any IR Rx data is erroneous (Active HIGH)
i_sys_clk	Input	100 kHz clock - Generated in top module
i_sys_rst	Input	System Reset - Connected to POR Logic
i_ir_in	Input	Input signal from IR receiver

The code starts with a state machine that detects whether data is being received. The state machine also indicates what type of data is being decoded/received. Each state of the state machine controls the data flow and the logic used to decode the data. The state machine also determines whether the incoming data is erroneous.

Various logic in this module are present for procedures such as edge detection of incoming data, counting the duration of received signals, and deserialization of the incoming data. The module data_decoder is specifically used to determine the duration of the received signals.

Lattice recommends that the IR RX Decoder not be changed.

SPI Interface to Application Processor (spi_reg)

This module is found in spi_reg file. It is used to interface between the iCE40LM Philips IR Rx Solution and the application processor. It also contains the data FIFO for IR Rx data and logic that issues interrupt to the application processor to indicate that IR Rx data is present for the processor to read. In addition to data FIFO, this module contains a SPI register called VERSION.

Table 7.2 summarizes the ports to/from this module.

Table 7.2. Ports To/From SPI Interface and Application Processor Module

Port Name	Direction	Description
i_clk	Input	System clock
i_rst	Input	System Reset - Connected to POR Logic
i_SPI_SCLK	Input	SPI interface (connected to i_SPI_SCLK)
i_SPI_SS_N	Input	SPI interface (connected to i_SPI_SS_N)
i_SPI_MOSI	Input	SPI interface (connected to i_SPI_MOSI)
o_SPI_MISO	Output	SPI interface (connected to o_SPI_MISO)
o_intr	Output	Interrupt to Application Processor - It signals the processor that IR Rx data is present and ready for read when data FIFO is not empty and when processor is not selecting this solution (Active HIGH) – Connected to o_sys_intr
o_fifo_rden	Output	Used to read the IR Rx Data FIFO (Active HIGH) - used only internal to this module
o_miso_byte_req	Output	Determines whether the received command is write (Active HIGH) or read (Active LOW)
o_test_empty	Output	Test Pin – Not Used
o_test_busy	Output	Test Pin – Not Used
i_RC6_control_field[7:0]	Input	IR Rx Control Field Data from RC6 Decoder
i_RC6_info_field[7:0]	Input	IR Rx Info Field Data from RC6 Decoder
i_RC6_mode[2:0]	Input	IR Rx Info Mode Data from RC6 Decoder
i_RC6_data_error	Input	Determines whether any IR Rx data is erroneous (Active HIGH)
i_RC6_data_vld	Input	Determines whether the IR Rx data is valid (Active HIGH)

The following is a walkthrough of the spi_reg code. Codes in this section are taken directly from the HDL file. Note that in most cases, the topics in each paragraph below are presented in the order in which they appear in the HDL code.

The first parameters defined in the code are: VERSION (which is for VERSION register), and DELAY_100_MICRO_SECOND (which is used for timeout for interrupt to processor (i.e. o_sys_intr)).

The code then proceeds to process the information received from application processor ("mosi_byte"). Mosi_byte is then decoded into command information: read and register address. Write instruction is not processed as there is nothing to write into.

When a read command is received:

- If reg_address = 0, then the VERSION register is sent to the application processor.
- If reg_address = 2, then the IR Rx Data FIFO are read and that data is sent to the application processor. Note that three consecutive reads must be performed in order to obtain mode, control, and info data recovered from the IR receiver module.

Recall that this module contains the FIFO to store data from IR Rx Decoder. The code then proceeds for control of this FIFO, starting from the section with comments "// fifo write data". Here, the three sets of data (mode, control, and info) are latched if and only if the data is valid. The same valid signal is then captured and delayed so as to enable multiple writes into the FIFO. Using this delayed valid signal, the FIFO is then written three times for the following data (in order of writes): mode, control, and info. Once the FIFO is not empty and that the chip select (i_SPI_SS_N) is not asserted, the logic issues an interrupt to the processor to indicate that data is present for the application processor. Also, notice that there is a interrupt timeout counter in the event that the interrupt is asserted too long.

At the end of the SPI Interface Module code, the spi_slave and a_fifo8 are instantiated.

Table 7.3 summarizes the ports to/from the a_fifo8 module

Table 7.3. Ports To/From the FIFO Module

Port Name	Direction	Description
i_rst	Input	Reset
i_wren	Input	Write Enable
i_wrclk	Input	Write Clock
i_wrdata[7:0]	Input	Write Data
i_rden	Input	Read Enable
i_rdclk	Input	Read Clock
o_rddata[7:0]	Output	Read Data
o_full	Output	Full
o_empty	Output	Empty

Table 7.4 summarizes the ports to/from the spi_slave module.

Table 7.4. Ports To/From the spi_slave Module

Signal	Direction	Description
i_sys_clk	Input	System Clock
i_sys_rst	Input	System Reset - Connected to POR Logic
i_miso_byte[7:0]	Input	Data to send to Application Processor
i_miso_byte_valid	Input	Determines if data to send is valid
o_miso_byte_req	Output	Determines whether the received command is write (Active HIGH) or read (Active LOW)
o_mosi_byte[7:0]	Output	Data received from Application Processor
o_mosi_byte_valid	Output	Determines if received data is valid (Active HIGH)
o_cmd_byte	Output	Determines if received data is a command byte (Active HIGH)
o_miso	Output	SPI interface (connected to proc_sdi pin)
i_mosi	Input	SPI interface (connected to proc_sdo pin)
i_csn	Input	SPI interface (connected to proc_csn pin)
i_sclk	Input	SPI interface (connected to proc_sclk pin)

The spi_slave module contains the hard SPI module called "SB_SPI". It contains logic that determines whether the command is write or read, and state machine to process the SPI master commands so as to prepare data for the backend interface.

8. Design Considerations

8.1. SPI Interface

This section describes the SPI interface between iCE40LM Philips IR Rx Solution and the Application Processor.

The Application Processor obtains IR Rx data over SPI lines through the spi_reg module. This module expects SPI in mode "3" format, i.e. CPHA = 1 and CPOL = 1, and MSB first while transmitting a byte of data over the bus.

The first byte after chip select assertion is treated as command byte, which would give the address of the register to-be-accessed. A dummy byte is sent in case of processor read operation to allow the read logic to decode the command and provide appropriate data in successive bytes. Note that the write operation is currently not doing anything.

The following timing diagrams show various read access patterns. Multi byte transaction is supported only for reading IR Rx data operation.

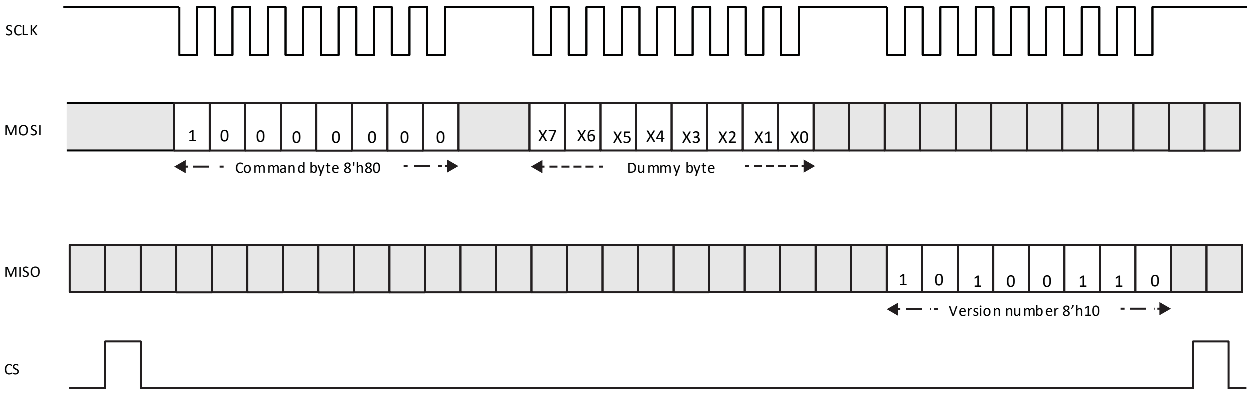


Figure 8.1. Timing Diagram of SPI Interface - Version Register Read

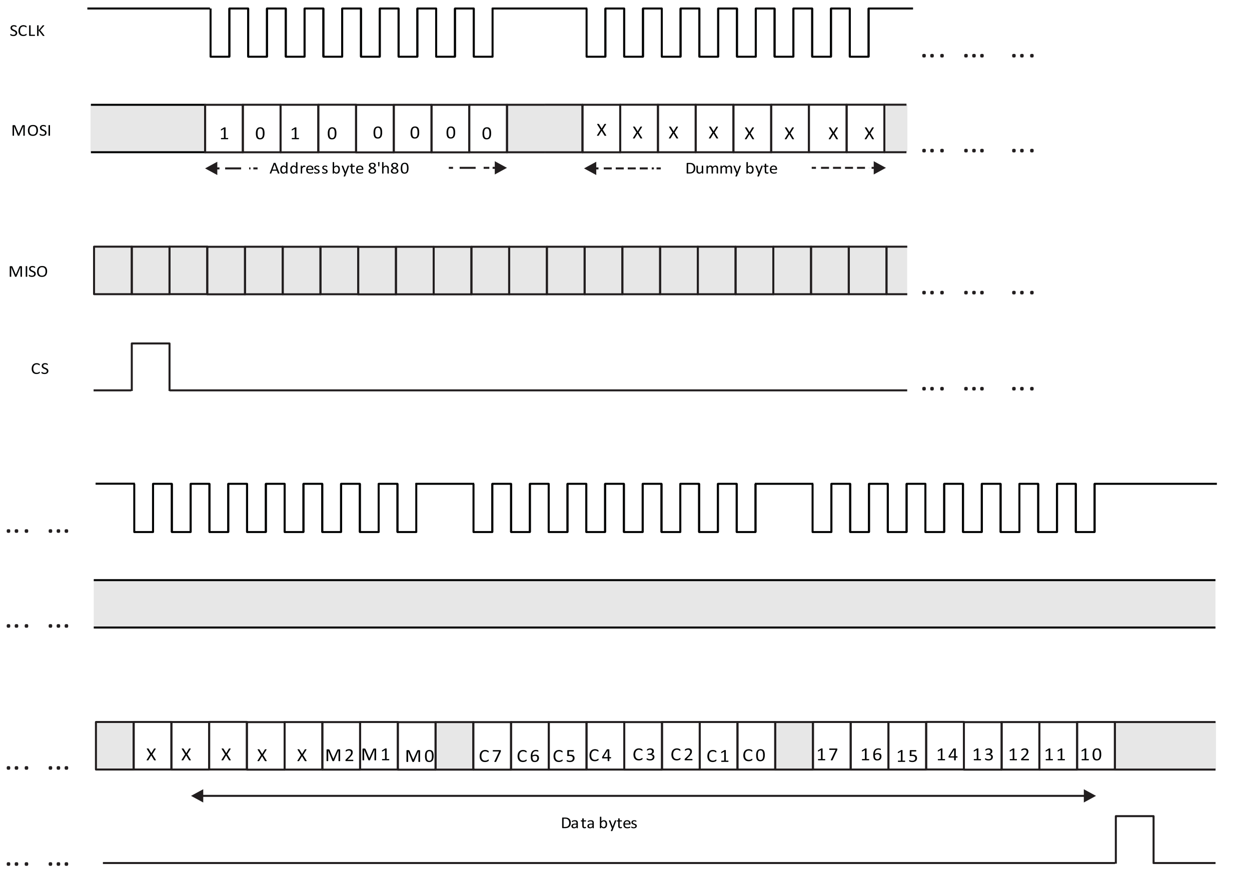


Figure 8.2. Timing Diagram of SPI Interface – IR Rx Data Read

Notes:

- For a read operation from processor, MSB of command byte is always 1.
- For a write operation from processor, MSB of command byte is always 0. Currently, no registers in the solution has write access.
- In the above timing diagrams, bits 6 to 4 under the Command byte indicate the register number in which read is applicable. Valid values are “000” and “010”.
- Bits 3 to 0 of command byte are not used.
- IR Rx Data Read is a multi-byte read operation.
- During IR Rx Data Read, only bits 2 to 0 of the first byte (mode) contain valid data.
- CS must not be asserted until all the bytes are read in case of multiple bytes read

8.2. SPI Registers Description

These registers are accessed by bits 6 to 3 of the Command byte. The following table describes these registers.

Table 8.1. Register Map for Bits 6 to 4 of Command Byte

Address (N2,N1,N2 as 3-bit hex)	Register Name	Access Type	Description
0x00	VERSION	R	Indicates the firmware version
0x02	DATA	R	IR Rx data register

Note:

1. Sensor specific registers requires N2, N1, and N0 bits of the Command byte to access.

Table 8.2. VERSION Register Bit Description

7	6	5	4	3	2	1	0
Firmware Version							

Table 8.3. DATA Register Bit Description

7	6	5	4	3	2	1	0
IR Rx Data [7:0]							

Data reading is multi-byte read operation (with single address). While reading the IR Rx data it is expected that the application processor is aware of the numbers bytes to be read for a particular device. Recovered data from IR Receiver Module is always a set of three bytes in the following order: mode, control, and info. Note that the first byte (mode) only has valid data at bits 2 to 0.

8.3. Complete SPI Registers Location

The table below lists the first byte to be transmitted from SPI master (AP) to iCE on MOSI Line. This is combination of register address listed in SPI Registers Description section and also the control signal values listed after the SPI Timing diagram (under Notes).

Table 8.4. First Byte from SPI Master (AP) to iCE40 on MOSI Line

First Byte for SPI Read	First Byte for SPI Write	Register Description
0x80	-	VERSION
0xA0	-	IR Rx Data

Example:

IR Rx Solution generates interrupt when IR received data is available in the FIFO. When data is available, IR Rx Solution interrupts the application processor by generating a high on "o_sys_intr" pin. Processor must follow the below mentioned steps to read the data from iCE.

1. When "o_sys_intr" goes high, read the data register (0xA0) by writing 0xA0 as first byte on i_SPI_MOSI (MOSI) line.
2. Read from the same register two more times to get the complete set of data.

8.4. Pseudo Code Example for Application Processor

The following code illustrates how an Application Processor could process the interrupt received from the IR Rx Solution to obtain the IR received data set.

```
Read IR Rx Data (0xA0)
```

8.5. Design Customization Considerations

Since this is an FPGA based solution, user can customize this solution by changing the source code of the IR Rx solution or add additional functions to this solution. Note that when customization is performed, the Performance Characteristics values might change.

8.6. Programming Solutions

Due to the FPGA nature of this solution, the solution requires FPGA programming. The programming solutions include, but not limited to programming via FTDI chip, programming via SPI Flash, or programming via application processor. For more information on programming solutions, please refer to "iCE40 Configuration Solutions Guide".

8.7. Power Supplies

Please refer to FPGA board design guide.

8.8. Layout Guidelines

Please refer to FPGA board design guide.

8.9. Heatsink Selection

Please refer to FPGA board design guide.

9. Software Requirement

For standalone solution, Diamond Programmer and “SPI_RC6_bitmap.hex” file. The following steps are required to program the device:

1. Create a new project
2. Set to SPI Programming

For fully customizable solution, iCEcube2, Diamond Programmer, and IR Rx HDL source files are required. For more information on iCEcube2, please refer to the iCEcube2 webpage.

10. Resource Utilization

LUTs	Registers	PLBs	BRAMs	I/Os	I2Cs	SPIs
418	256	77	1	7	0	1

11. Typical Application Circuits

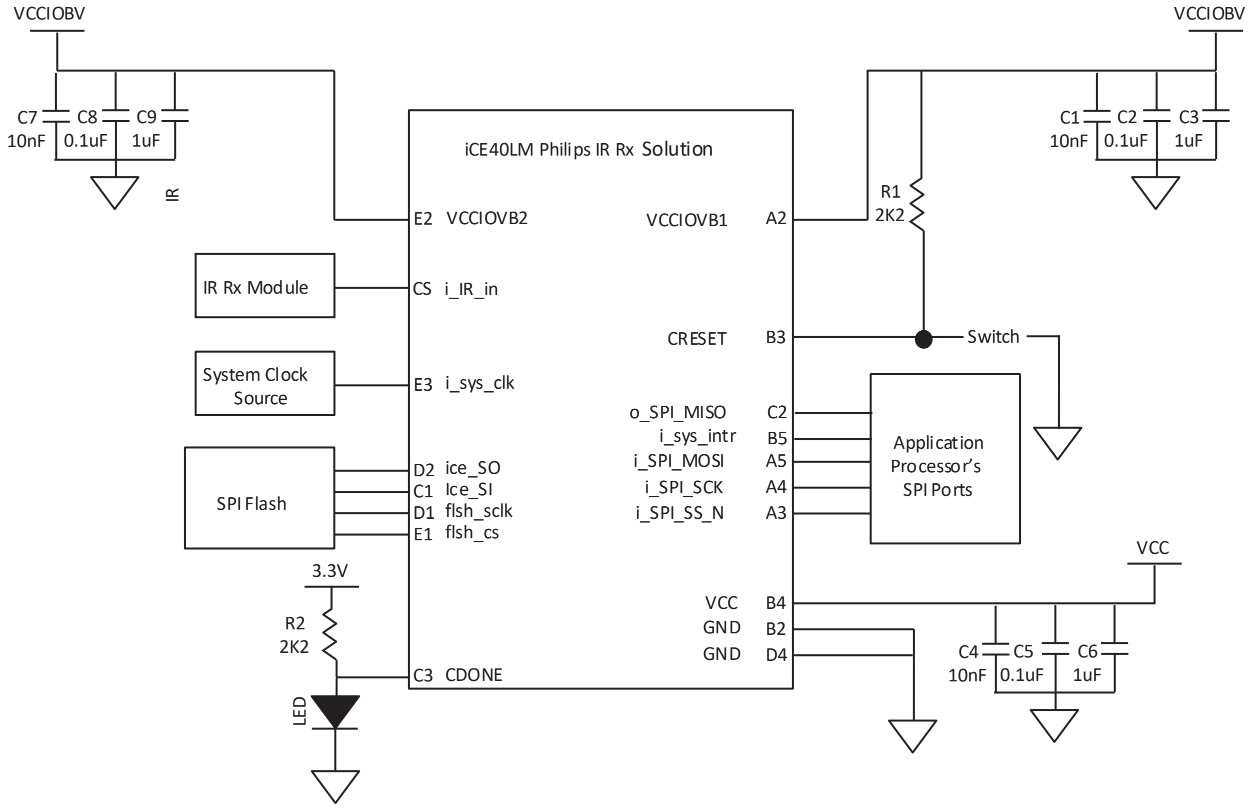


Figure 11.1. IR Rx with Pre-programmed SPI Flash

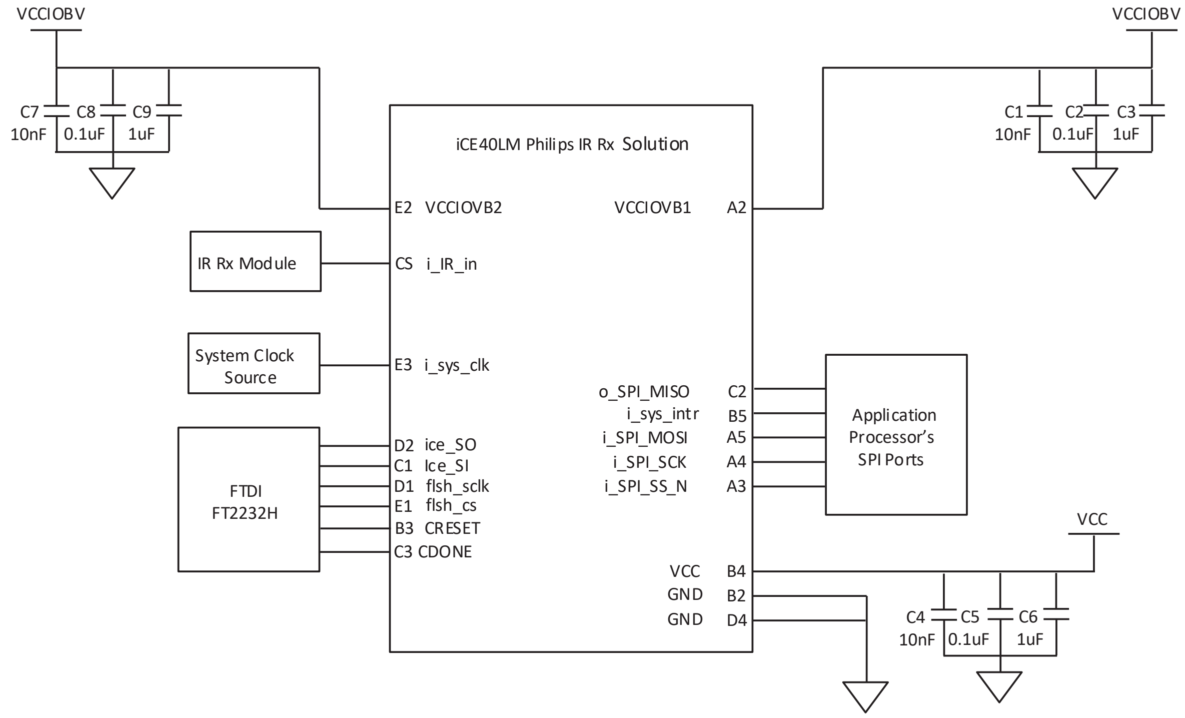


Figure 11.2. IR Rx with Direct Programming through FTDI

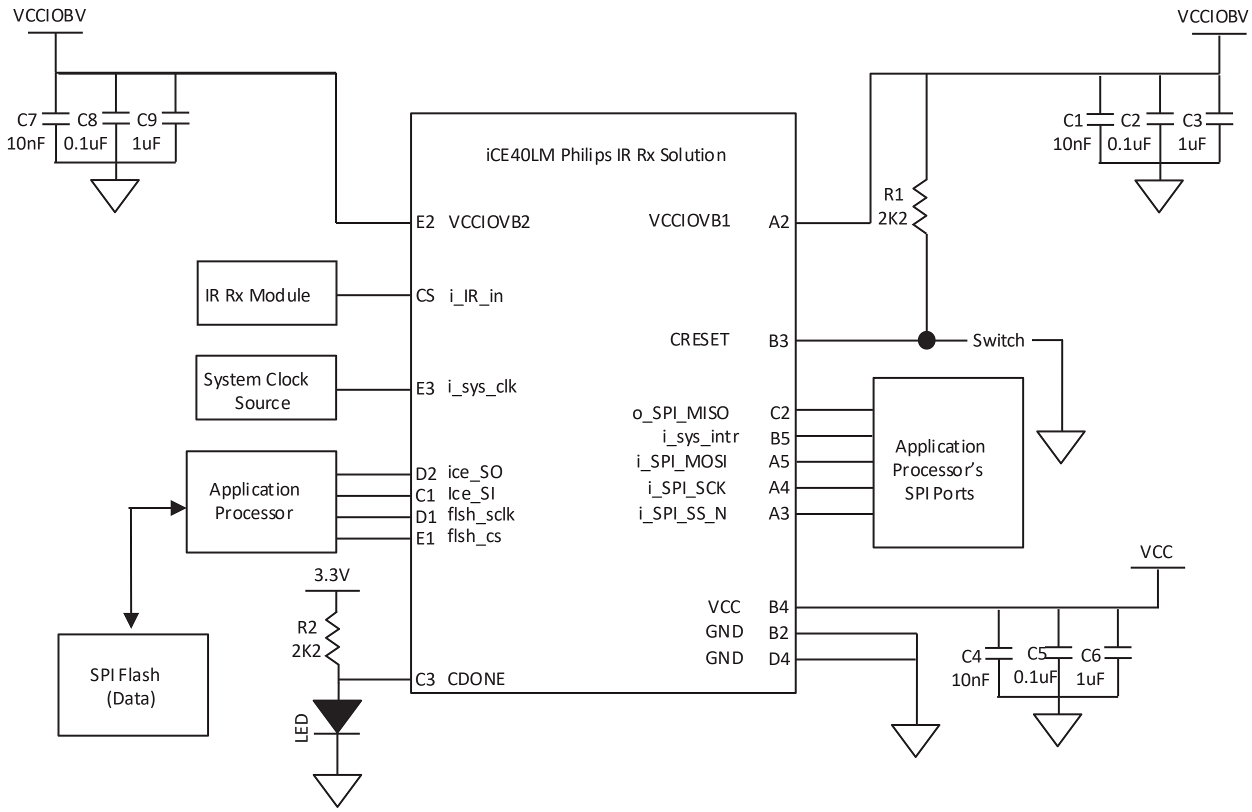
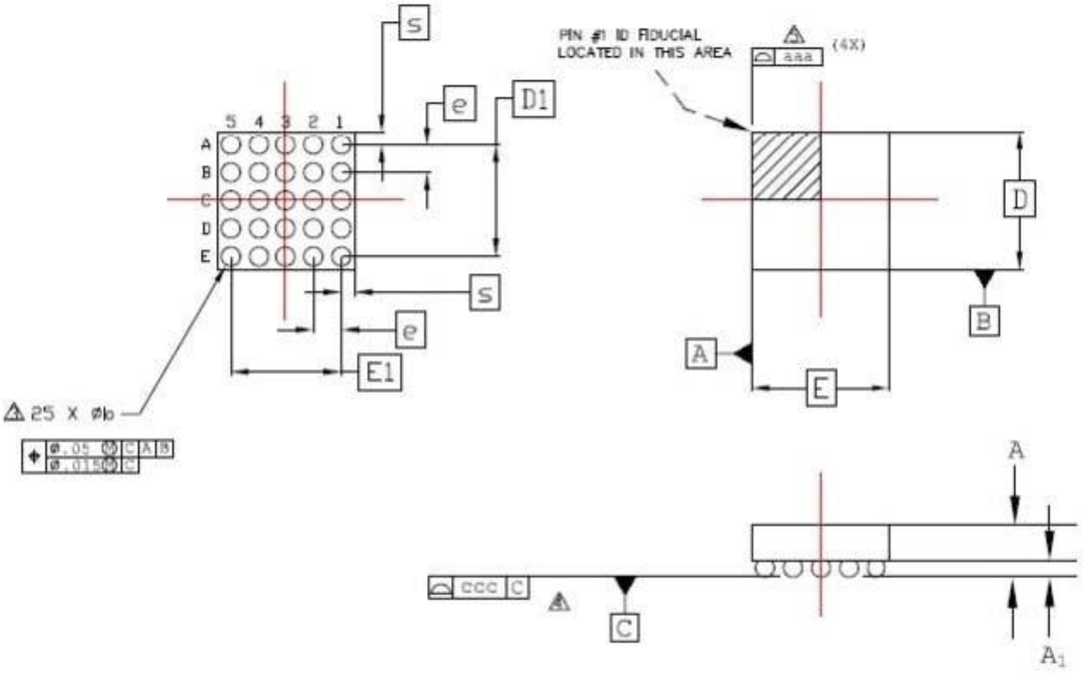


Figure 11.3. IR Tx with Programming through Application Processor

12. Package Diagram



Notes:
1 ALL DIMENSIONS AND TOLERANCE PER ASME Y 14.5M - 1994.
2 ALL DIMENSIONS ARE IN MILLIMETERS.
△ DIMENSION "b" IS MEASURED AT THE MAXIMUM BUMP DIAMETER PARALLEL TO PRIMARY DATUM C.
△ PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BUMPS.
△ BILATERAL TOLERANCE ZONE IS APPLIED TO EACH SIDE OF THE PACKAGE BODY.

REF.	Min.	Nom.	Max.
A	0.413	0.452	0.491
A1	0.122	0.152	0.182
b	0.188	0.218	0.248
D		1.71 BSC	
E		1.71 BSC	
D1		1.40 BSC	
E1		1.40 BSC	
e		0.35 BSC	
aaa		0.03	
ccc		0.03	
s	–	0.155	–

13. Disclosures

The ICE40LM Philips IR Rx Solution is an FPGA based solution which requires IP to be downloaded to the device for this solution. This solution includes the Diamond Programmer for IP download and iCEcube2 design software for customization. The design files and ready-for-download .hex file are also included. Finally, SPI Flash might be needed depending on whether one time or multi programmable scheme is used.

14. Ordering Information

Solution Name	Description	Package	BOM
iCE40LM Philips IR Rx Solution (Commercial Grade)	Commercial Grade Solution	25-pin WLCS at 1.71 mm x 1.71 mm	iCE40LM4K-SWG25TR Device, iCEcube2 Design Software, Diamond Programmer, IR Tx Design Files, SPI_RC6_bitmap.hex
iCE40LM Philips IR Rx Solution (Industrial Grade)	Industrial Grade Solution	25-pin WLCS at 1.71 mm x 1.71 mm	iCE40LM4K-SWG25TR Device, iCEcube2 Design Software, Diamond Programmer, Sony IR Tx Design Files, SPI_RC6_bitmap.hex

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.2, February 2020

Section	Change Summary
All	• Changed document number from RD1192 to FPGA-RD-02159. • Updated document template.
Disclaimers	Added this section.

Revision 1.1, October 2013

Section	Change Summary
Design Considerations	• Updated the Timing Diagram of SPI Interface - Version Register Read figure. • Updated the Timing Diagram of SPI Interface – IR Rx Data Read figure.

Revision 1.0, October 2013

Section	Change Summary
All	Initial release.

LATTICE SEMICONDUCTOR®

www.latticesemi.com