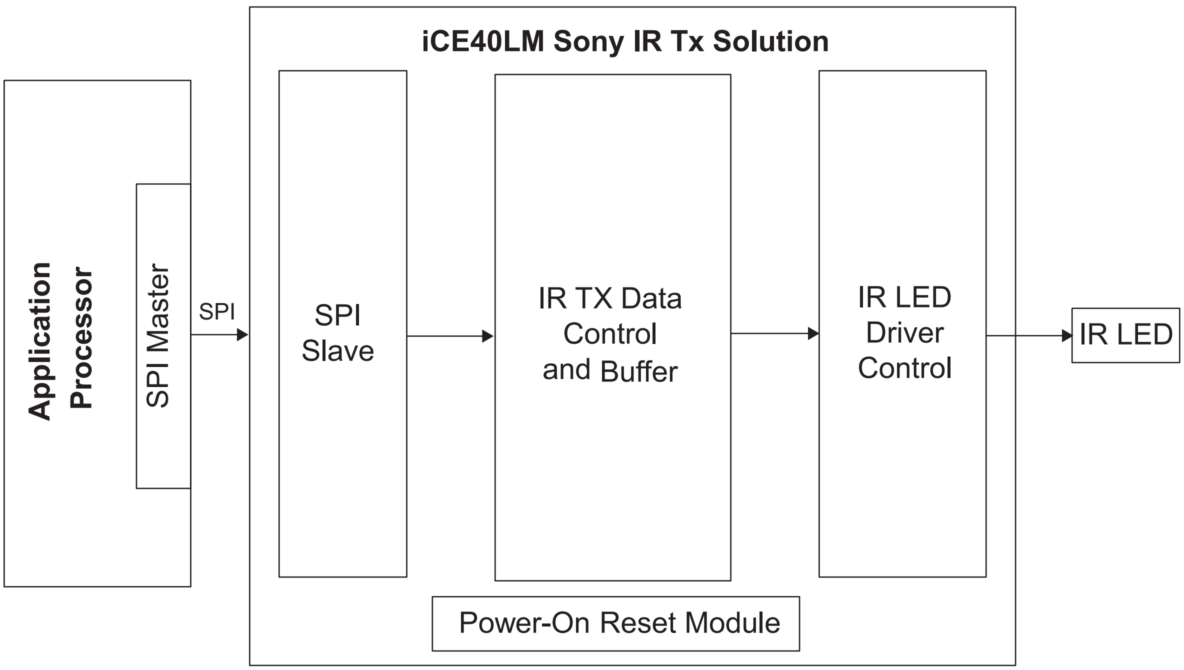
iCE40LM Sony IR Tx Solution

October 2013 Reference Design RD1190

General Description

The iCE40LM Sony IR Tx Solution is a low power IR Tx solution for mobile devices. It is designed to easily send IR Tx data from the application processor to the IR LED. This reference design acts as data control and buffer between the IR LED and the application processor. The iCE40LM Sony IR Tx Solution is a configurable solution, available as either standalone off the shelf or fully customizable solution, making it an IR Tx standard agnostic solution.

Figure 1. System Block Diagram



As a standalone solution, the iCE40LM Sony IR Tx Solution connects to the application processor's Serial Peripheral Interface Bus (SPI) with clock frequency set to 10.8MHz. This enables a fast communication speed to/from the processor. The iCE40LM Sony IR Tx standalone solution prepares and sends the data from the application processor to an IR LED. The data is sent to the IR LED with the correct frequency, duration, and interval as required by Sony IR Tx standard.

The iCE40LM Sony IR Tx standalone solution has a system operating frequency of 27MHz, and a SPI bus frequency to application processor of 10.8MHz. The SPI bus is configured to have a voltage of 1.8V, and the IR LED is driven by a 3.3V I/O.

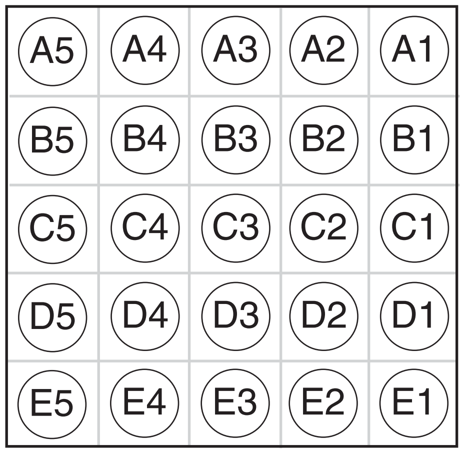
The fully customizable solution capability of this solution is due to its FPGA based architecture. This capability is ideal, but not limited to users who would like to include additional IR Tx standards, change the data acquisition FIFO depth, create an I/O bridge between IR LEDs and processor, or create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

The IR Tx reference design consumes only 335 LUTs. This allows it to fit in a device as small as a iCE40LM1K.

Regardless of the solution type, the iCE40LM implementation of the IR Tx Solution has a core voltage of 1.2V. It is available in a very small form-factor 25-pin WLCSP package. The package has 0.35mm ball pitch, making the overall package size to be 1.71mm x 1.71mm that easily fit into a number of mobile devices such as smart phones. Other packages include .4mm ball pitch with 36 balls (2.5x2.5mm) or 49 balls (3x3mm). The solution operates at industrial temperature range of -40C to 100C.

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 Sony IR Tx solution.

Figure 2. Package Diagram (Balls Up)



Features

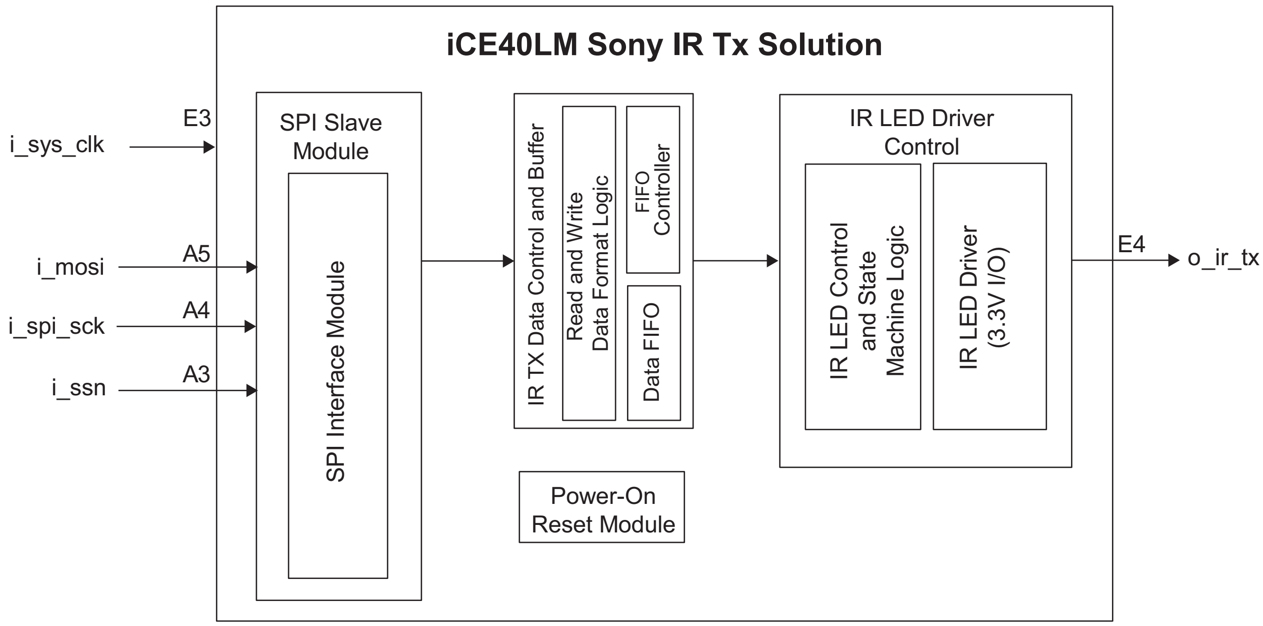
- Configurable IR Tx Solution
  - Configured to Sony IR Tx Interface
  - Default System frequency of 27MHz
  - Power-On Reset capability
- Serial Peripheral Interface (SPI) Bus connection to Application Processor with the following Default settings:
  - Interface frequency of 10.8MHz
  - Interface voltage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

Applications

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

Functional Block Diagram

Figure 3. Functional Block Diagram



Specifications

Recommended Operating Conditions

Table 1. Recommended Operating Conditions

Symbol	Parameter	Min.	Typ.	Max.	Units
V_{CC}	Core Supply Voltage	See DS1045, iCE40LM Family Data Sheet.	1.2	See DS1045, iCE40LM Family Data Sheet.	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 DS1045, iCE40LM Family Data Sheet.	-	See DS1045, iCE40LM Family Data Sheet.	°C

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

Power Supply Ramp Rates

See DS1045, iCE40LM Family Data Sheet.

Power-On-Reset Voltage Levels

See DS1045, iCE40LM Family Data Sheet.

ESD Performance

See DS1045, iCE40LM Family Data Sheet.

DC Electrical Characteristics

See DS1045, iCE40LM Family Data Sheet. Set the $_{VCCIO}$ values to the values stated in the Recommended Operating Conditions table.

Power Supply Current

See DS1045, iCE40LM Family Data Sheet.

Absolute Maximum Ratings

See DS1045, iCE40LM Family Data Sheet.

Performance Characteristics

Table 2. Performance Characteristics[1, 3]

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

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.

FPGA Characteristics

See DS1045, iCE40LM Family Data Sheet. Note that once customization is performed, the values in "Performance Characteristics" may not be the same.

Pin Configuration and Function Descriptions

Figure 4. Bottom View of iCE40LM4K-SWG25TR (Balls Up)

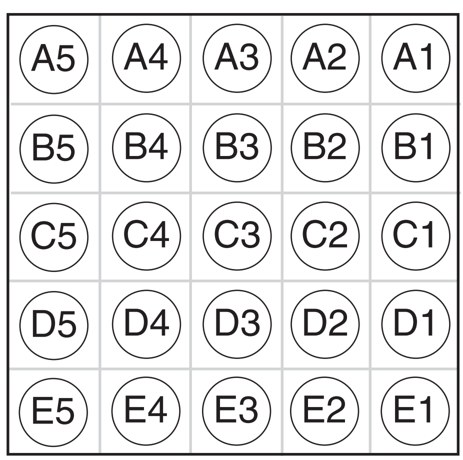


Table 3. Pin Function Description[1]

Pad Name	Port Name	Port Direction	Description
A1	General Purpose I/O	Input/Output	1.8V I/O for user interface
A2	VCCIOVB1	Input	I/O Power Supply
A3	i_ssn	Input	SPI bus slave select (Active Low)
A4	i_spi_sck	Input	SPI bus serial clock
A5	i_mosi	Input	SPI bus serial data in to slave
B1	General Purpose I/O	Input/Output	1.8V 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	General Purpose I/O	Input/Output	1.8V I/O for user interface
C1	ice_SI	Output	Configuration Output to external SPI Memory
C2	General Purpose I/O	Input/Output	1.8V I/O for user interface
C3	CDONE	Output	Configuration Done. See Datasheet
C4	General Purpose I/O	Input/Output	3.3V I/O for user interface
C5	General Purpose I/O	Input/Output	3.3V I/O for user interface
D1	flsh_sclk	Input	Configuration Clock
D2	ice_SO	Input	Configuration Input from external SPI Memory
D3	General Purpose I/O	Input/Output	3.3V I/O for user interface
D4	GND	Input	Ground
D5	General Purpose I/O	Input/Output	3.3V 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	o_ir_tx	Output	IR LED Output Driver (3.3V)
E5	General Purpose I/O	Input/Output	3.3V I/O for user interface

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

Theory of Operations

The iCE40LM Sony IR Tx Solution interfaces between an application processor and an IR LED. It receives Sony based IR Tx data from the processor through the SPI bus. The received data is then formatted for IR driver logic and stored/buffered into a data FIFO. Once the data FIFO is no longer empty, the IR driver logic reads the FIFO content and converts the data into Sony compatible 3.3V 40kHz serial signals that drives the IR LED. It continues to drive the LED until the FIFO is empty, and the whole process begins again when the next set of received data is present.

Functional Descriptions

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

Sony IR Tx Top Level

The Sony IR Tx Top Level is found in Top_level. This module contains the SPI Slave to Application Processor, the IR Tx Data Control and Buffer, and IR LED Driver Control. It also contains a Power-On Reset (POR) module. The POR module initiates a system reset upon power up for Tpor number of cycles. The iCE40LM Sony IR Tx Solution operates after system reset has been completed.

SPI Slave to Application Processor

This module is used to interface between the iCE40LM Sony IR Tx Solution and the application processor. It is found in SPI_Slave_Wrapper module. This module can only receive data from application processor. Logic to send data to the application processor is not implemented. When data is sent from the application processor, this module simply converts the data from SPI serial data to 8-bit data and indicates whether that data is valid. Note that a set of Sony IR Tx data is expected to come into this solution as a set of three consecutive 8-bit data.

IR Tx Data Control and Buffer

This module takes the three 8-bit data from the SPI Slave and formats the three bytes into a 21-bit word. After the formatting is completed, the 21-bit word is written into a FIFO. When the FIFO is not empty, this module issues a signal to the IR LED Driver Control module that data is present in the FIFO. The FIFO content is then read and prepared for the IR LED Driver Control module. The information sent to the IR LED Driver Control module is: 20-bit data, 1-bit to indicate whether the data is 20 or 12 bits, and 1-bit to indicate whether the data is valid.

IR LED Driver Control

This module issues a FIFO read to the IR Tx Data Control and Buffer module when the FIFO from that module is not empty. The signals received from the IR Tx Data Control and Buffer module are then used to calculate the duration of each bit that needs to be transmitted. This module then generates a 40kHz with duty cycle of approximately 25% to drive the IR LED. Controlled by a state machine, each bit is then sent serially via 3.3V output using the 40kHz as the carrier signal with the appropriate durations. The bits are sent from LSB to MSB.

Block Descriptions

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

Top Level Module (Top_level)

This module contains the SPI Slave to Application Processor, the IR Tx Data Control and Buffer, and IR LED Driver Control. The HDL code begins with the POR logic, which is set to Tpor cycles, and the three sub-modules above are instantiated and connected together.

The code SPI_Slave_wrapper instantiation is the SPI Slave to Application Processor module. This module can only receive data from the application processor. The received data is then sent to the IR Tx Data Control and Buffer module (SPI_Slave_Registers) where it is formatted for the IR LED Driver Control and stored in FIFO. When that FIFO is no longer empty, the IR LED Driver Control module reads the content of the FIFO. The IR LED Driver Control module then sends out the data serially with the appropriate duration.

SPI Slave to Application Processor Module (SPI_Slave_wrapper)

This module is found in the SPI_Slave_wrapper file. It is used to provide connection from the application processor to this solution via SPI interface. Note that in this solution, user can only send data from the application processor to the solution, not the other way around.

Table 4 summarizes the ports to/from this module.

Table 4. Ports To/From Application Processor Module

Port Name	Direction	Description
o_data[7:0]	Output	Data received from Application Processor
o_tx_ready	Output	Unused
o_rx_ready	Output	Determines if received data is valid (Active HIGH)
o_tx_error	Output	Unused
o_rx_error	Output	Unused
o_miso	Output	Unused
o_tx_ack	Output	Unused
o_tx_no_ack	Output	Unused
i_sys_clk	Input	System Clock
i_sys_rest	Input	System Reset - Connected to POR
i_csn	Input	Unused
i_data[15:0]	Input	Unused
i_wr	Input	Unused
i_rd	Input	Unused
i_cpol	Input	Unused
i_cpha	Input	Unused
i_lsb_first	Input	Unused
i_mosi	Input	SPI serial data into slave - connected to I/O
i_ssn	Input	SPI slave select (Active Low) - connected to I/O
i_sclk	Input	SPI clock input - connected to I/O

The SPI Slave to Application Processor Interface Module code simply instantiates the spi_slave module.

Table 5 summarizes the ports to/from the spi_slave module:

Table 5. Ports To/From spi_slave Module

Port Name	Direction	Description
o_miso_byte_req	Output	Unused
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	Unused
o_miso	Output	Unused
i_sys_clk	Input	System Clock
i_sys_rst	Input	System Reset - Connected to POR
i_miso_byte[7:0]	Input	Unused
i_miso_byte_valid	Input	Unused - Tied to Logic 1
i_mosi	Input	SPI serial data into slave - connected to I/O
i_csn	Input	SPI slave select (Active Low) - connected to I/O
i_sclk	Input	SPI clock input - connected to I/O

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

IR Tx Data Control Module (SPI_Slave_Registers)

This module is found in the SPI_Slave_Registers file. It receives data from the SPI Slave and formats them from three 8-bit data to one 21-bit data. The formatted data are then written into a FIFO. When the FIFO is not empty, this module issues a signal to the IR LED Driver Control module that data is present in the FIFO. The FIFO content is then read, and prepared for the IR LED Driver Control module.

The following table summarizes the ports to/from the IR Tx Data Control Module:

Table 6. Ports To/From IR Tx Data Control Module

Port Name	Direction	Description
o_sys_intr	Output	Indicates if o_sirc_word is valid data (Assert HIGH)
o_sirc_word[19:0]	Output	Data to send to the IR TX Controller
o_sirc_len	Output	Indicates length of o_sirc_word (1 = 20-bit, 0 = 12-bit)
o_fifo_empty	Output	Indicate if FIFO is empty (Assert HIGH) - Used to tell the IR TX controller to begin reading the FIFO so to issue command to remote device
i_sirc_fifo_wr_data[7:0]	Input	Data received from Application Processor
i_sirc_fifo_wr_en	Input	Determines if received data is valid (Active HIGH)
i_sirc_fifo_rd_en	Input	Indicate to read data received from Application Processor - Controlled by o_tx_read_req of Ir_Tx_Ctrl, which is controlled by FIFO Empty
i_sys_clk	Input	System Clock
i_sys_rst	Input	System Reset - Connected to POR

The following is a walkthrough of the SPI_Slave_Registers 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 set of logic in this module converts a set of three successive bytes of data from the application processor into 21-bit data. It assumes that the first two bytes, and bit 0 to 3 and 7 of the third byte contains data. There's a counter that keeps track of the number of bytes received in a set of data. The counter is also used to determine when the data is ready so that they can be written into the FIFO. The codes for this logic are locates under the following comments:

- // Three bytes of data from slave stored as 21 bits word
- // FIFO Write enable
- // Counter to keep track of three bytes from slave

Once the data has been formatted, they are written to a 4 deep by 21-bit FIFO.

The following table summarizes the ports to/from the FIFO:

Table 7. Ports To/From the FIFO

Port Name	Direction	Description
o_RdData[20:0]	Output	Data to send to the IR TX Controller
o_Full	Output	Indicate if FIFO is full (Assert HIGH) - Not used in solution
o_Empty	Output	Indicate if FIFO is empty (Assert HIGH) - Used to tell the IR TX controller to begin reading the FIFO so to issue command to remote device
i_clk	Input	System Clock
i_rst	Input	System Reset - Connected to POR
i_RdEn	Input	Read Enable for o_RdData
i_WrEn	Input	Write Enable for WrData
i_WrData[20:0]	Input	Data received from Application Processor (formatted to 21-bit)

After data has been written, the o_Empty will be deasserted, which will indicate the IR LED Driver Control (through o_fifo_empty) to issue a read command to the FIFO. When data has been read, they are now further refined for the IR LED Driver Control. They're formatted into one 20-bit data bus, a 1-bit signal to indicate whether the data is 20 or 12 bits, and 1-bit to indicate whether the data is valid. The logic for this operation is located under the following comments:

- // To generate delayed versions of fifo_rd_en pulses
- // To generate o_sirc_word and o_sirc_len after reading from fifo

The signals sent to the IR LED Driver Control Module are: o_sys_intr (data valid), o_sirc_word (20-bit data), and o_sirc_len (indicates whether o_sirc_word is 20 or 12-bit).

IR LED Driver Control Module (Ir_Tx_Ctrl)
This module is found in Ir_Tx_Ctrl file. It issues a FIFO read to the IR Tx Data Control and Buffer module when the FIFO from that module is not empty. The signals received from the IR Tx Data Control and Buffer module are then used to calculate the duration of each bit that needs to be transmitted. This module then generates a 40kHz with duty cycle of approximately 25% to drive the IR LED. Controlled by a state machine, each bit is then sent serially via 3.3V output using the 40kHz as the carrier signal with the appropriate durations. The bits are sent from LSB to MSB. This module has been designed specifically for Sony IR Tx interface.

The following table summarizes the ports to/from the IR LED Driver Control Module:

Table 8. Ports To/From IR LED Driver Control Module

Port Name	Direction	Description
o_tx_read_req	Output	Indicate to read data received from Application Processor - Controlled by FIFO Empty, and connect to SPI_Slave_Registers (i_sirc_fifo_rd_en)
o_ir_tx	Output	IR driver port
i_sys_clk	Input	System Clock
i_sys_rst	Input	System Reset - Connected to POR
i_fifo_empty	Input	Indicate if FIFO for data from processor is empty
i_sirc_word	Input	Data from SPI_Slave_Registers
i_sirc_len	Input	Indicate length i_sirc_word (1 = 20-bit, 0 = 12-bit)
i_read_data_vld	Input	Indicates if i_sirc_word is valid data (Assert HIGH)

The following is a walkthrough of the Ir_Tx_Ctrl 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.

When the FIFO at the IR Tx Data Control Module is not empty, this module generates a FIFO read signal if the IR LED Driver is available. The code is shown under the comment:

//Generates read request to FIFO of I2C Slave registers

Once data is received, they are prepared for transmit to the IR LED. The preparation includes: counting the number of 1s in the data, delaying the data until number of 1s is available, and time offset. These are located under:

- // Count number of 1s, used to compute how much of time left within 45ms
- //ones_count_i available afterone cycle and hence use this registered
- //wait_45ms_offset_i available after 1 cycles and hence use this registered

There's also code to generate the 40kHz signal with 25% duty cycle, per Sony IR Tx requirements. The 40kHz signal is only active whenever there's data to be sent. The codes are located under:

- //IR output active only in START_STATE and Data state, not 40KHz carrier wave in
- //IR Tx output, with 25% duty cycle

A state machine is used to control the logic above. The same state machine is used to control the serial data output to the IR LED. In addition, a number of counters are implemented to keep track of the time durations required by Sony IR Tx standard.

Design Considerations

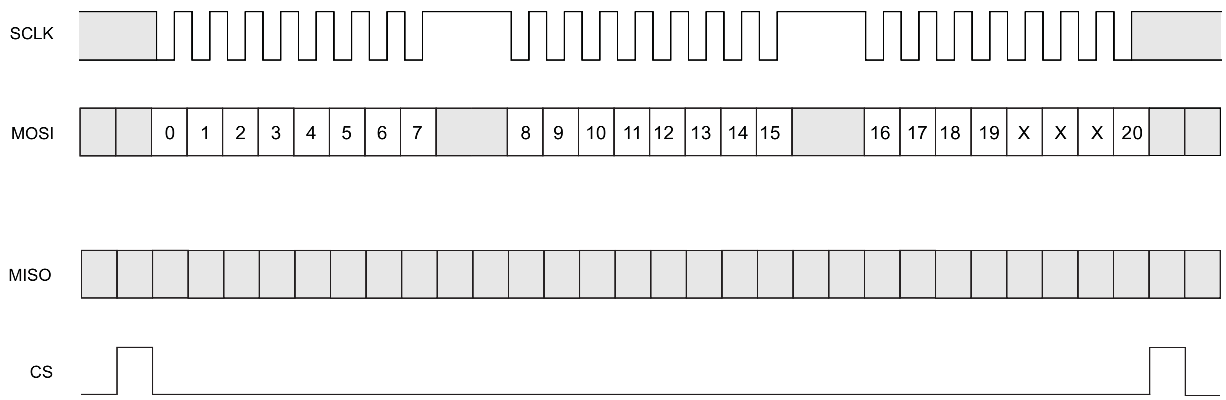
SPI Interface

This section describes the SPI Interface between iCE40LM Sony IR Tx Solution and the Application Processor.

The Application Processor sends data to the iCE40LM Sony IR Tx Solution over SPI lines through the SPI_Slave_wrapper module. This module expects SPI in mode “3” format, i.e. CPHA = 1 and CPOL = 1, and LSB first while transmitting a byte of data over the bus. Note that user can only perform a three-byte writes into this solution.

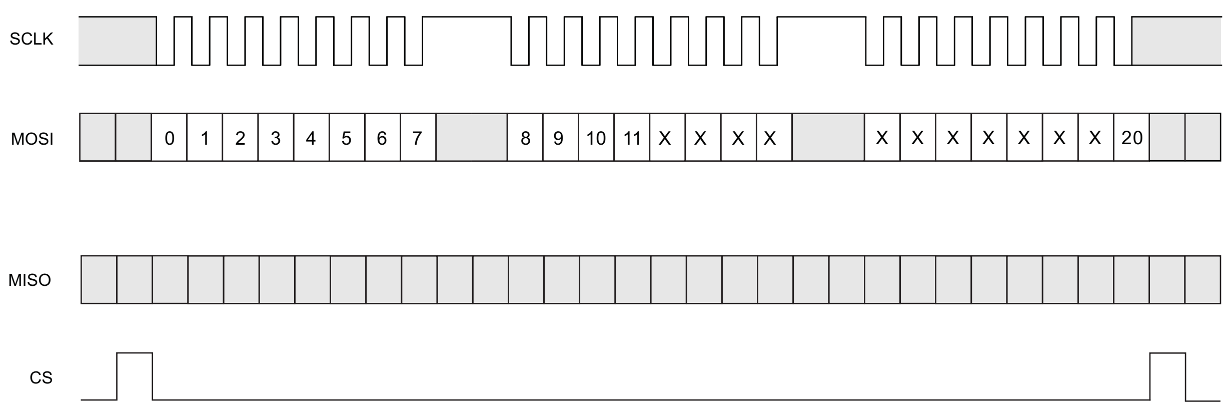
The following timing diagrams show the write access patterns.

Figure 5. Timing Diagram of SPI Interface - Multi Byte Write for 20-bit IR Tx Data



In Figure 5, bit 0 goes first and the last bit is the bit that indicates whether it is 20-bit or 12-bit data. “x” are “don’t cares”.

Figure 6. Timing Diagram of SPI Interface - Multi Byte Write for 12-bit IR Tx Data



In Figure 6, bit 0 goes first and the last bit is the bit that indicates whether it is 20-bit or 12-bit data. “x” are “don’t cares”.

Note: CS must not be asserted until all the bytes are read in case of multiple bytes read

Pseudo Code Example for Application Processor

The following code illustrates how an Application Processor could process the interrupt received from the IR Tx to obtain the sensor data.

```
Write three bytes of data to the iCE40LM Sony IR Tx Solution. Third data byte has
valid data at bit 0 to 3 and 7, where 7 has the data length indicator (1 = 20-bit,
0 = 12-bit).
```

Design Customization Considerations

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

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”.

Power Supplies

Please refer to FPGA board design guide.

Layout Guidelines

Please refer to FPGA board design guide.

Heatsink Selection

Please refer to FPGA board design guide.

Software Requirement

For standalone solution, Diamond Programmer and “Top_level_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 Tx HDL source files are required. For more information on iCEcube2, please refer to the iCEcube2 web page.

Resource Utilization

LUTs	Registers	PLBs	BRAMs	I/Os	I2Cs	SPIs
335	231	81	2	5	0	1

Typical Application Circuits

Figure 7. IR Tx with Pre-programmed SPI Flash

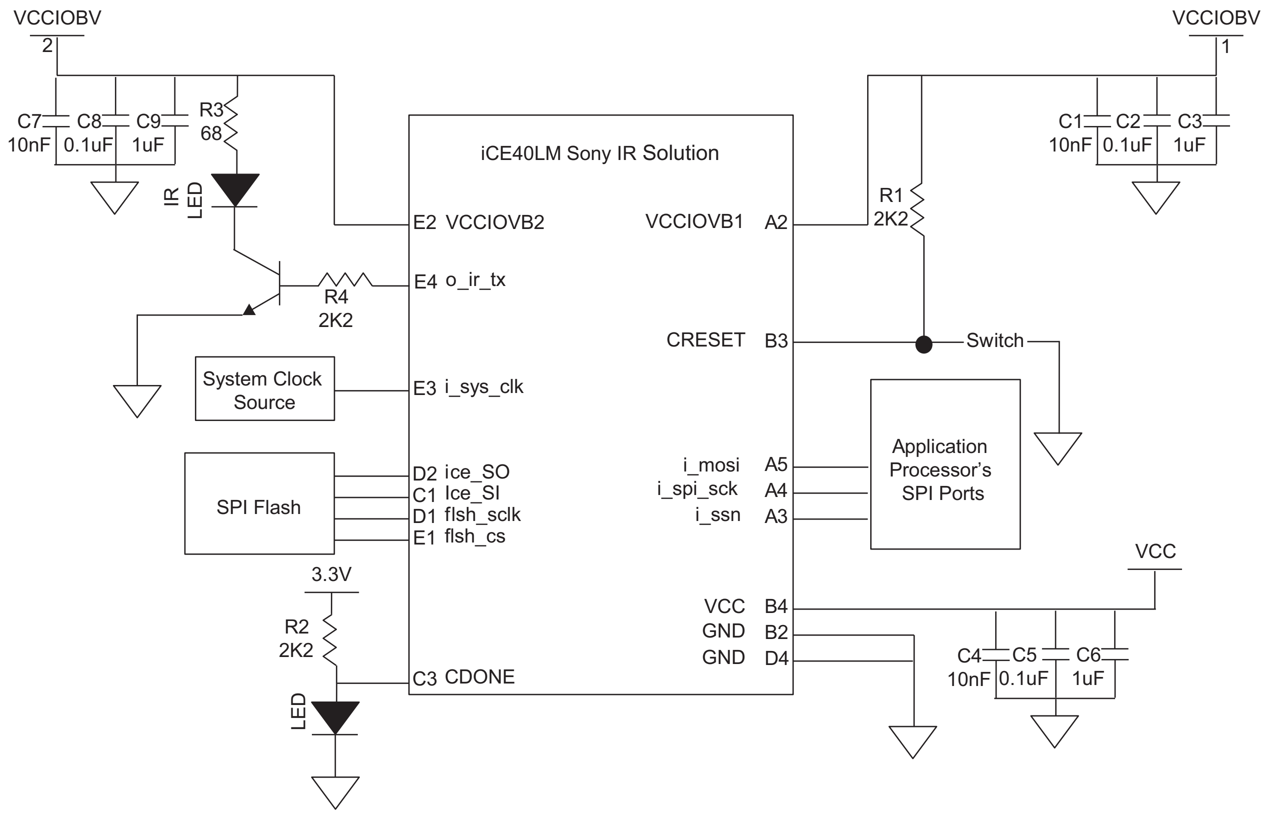


Figure 8. IR Tx with Direct Programming through FTDI

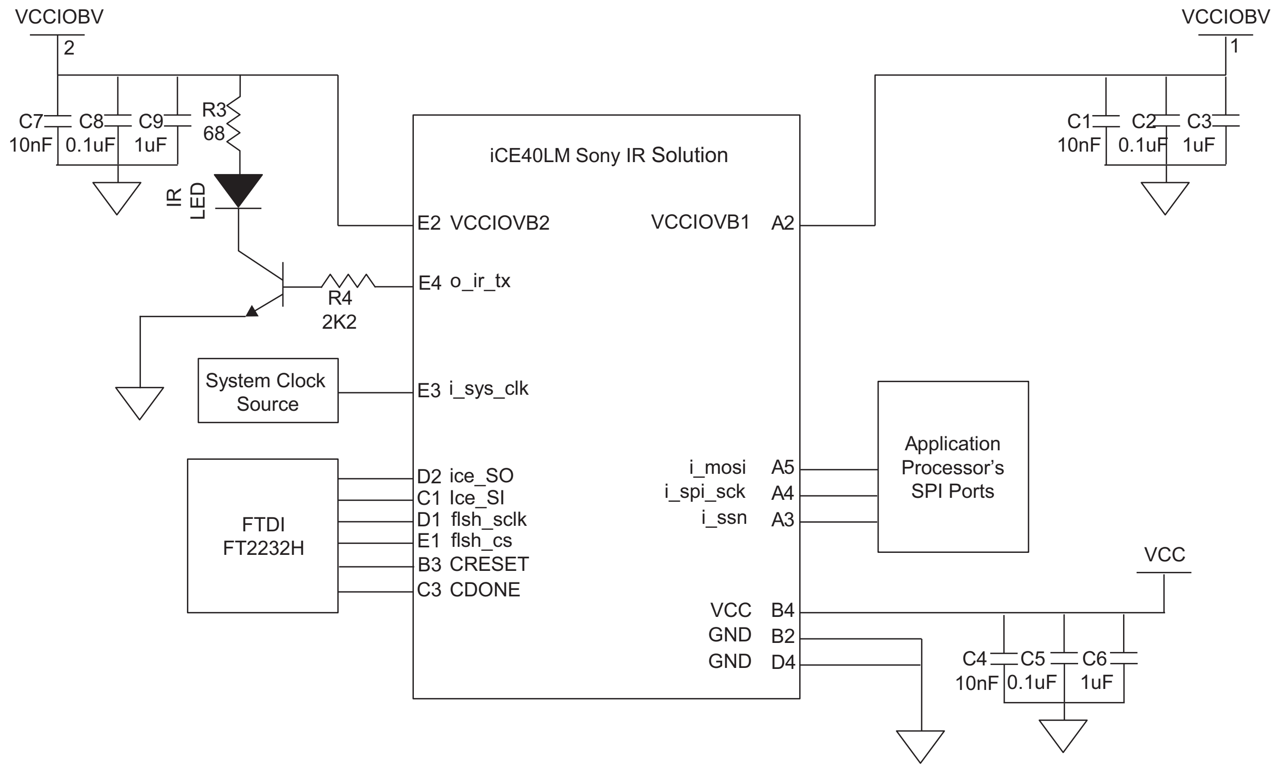
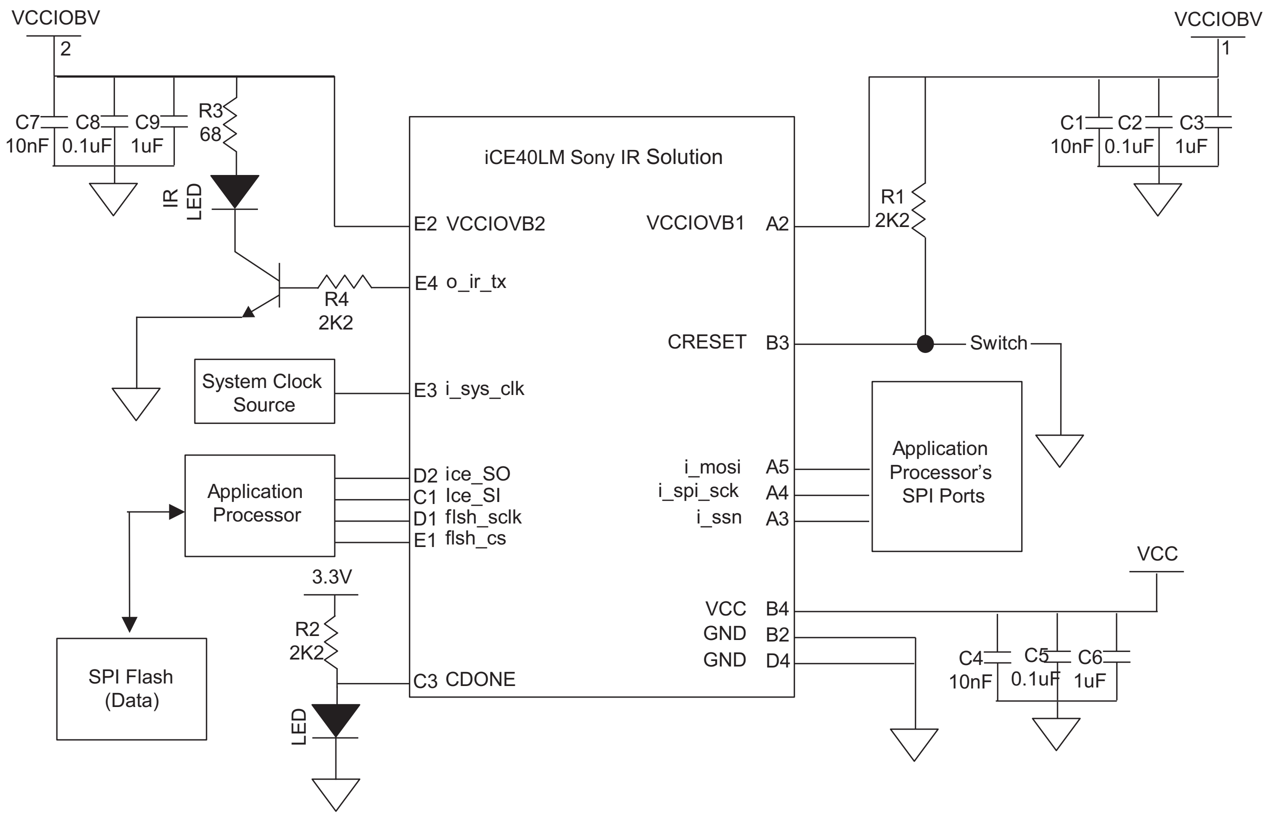
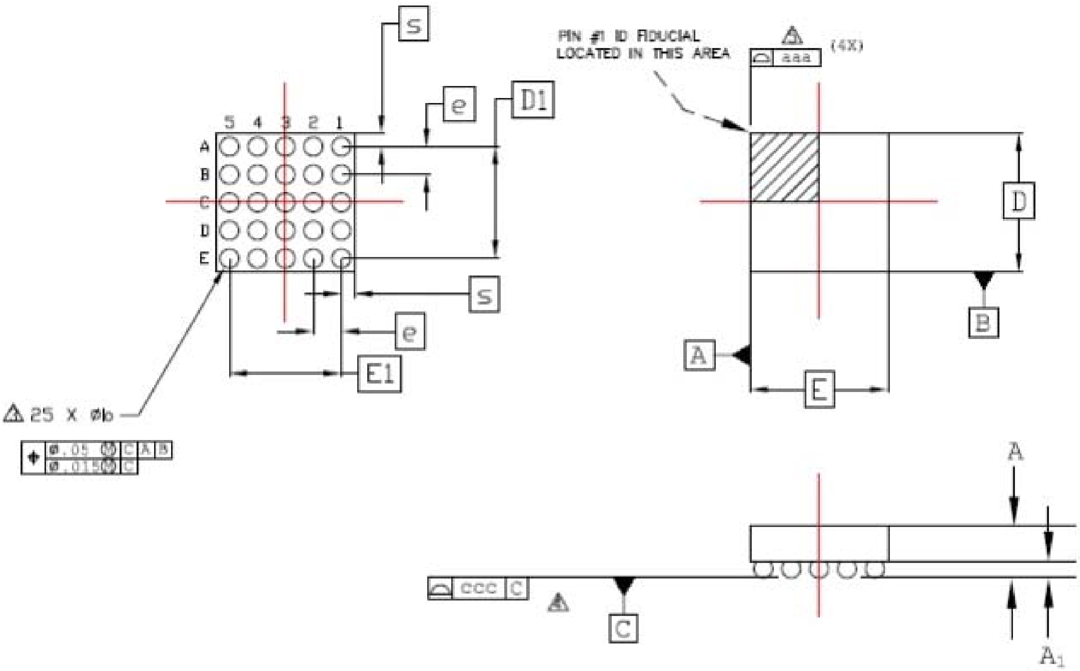


Figure 9. IR Tx with Programming through Application Processor



Package Diagram



Notes:
1 ALL DIMENSIONS AND TOLERANCE PER ASME Y 14.5M - 1994.
2 ALL DIMENSIONS ARE IN MILLIMETERS.
3 DIMENSION 'b' IS MEASURED AT THE MAXIMUM BUMP DIAMETER PARALLEL TO PRIMARY DATUM C.
4 PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BUMPS.
5 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	-

Disclosures

The iCE40LM Sony IR Tx 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.

Ordering Information

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

Technical Support Assistance

e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
October 2013	01.0	Initial release.
	01.1	Changed system operating frequency of 25MHz to 27MHz.
		Updated SPI Interface section
		Updated Pseudo Code Example for Application Processor section.