Lattice Single Wire Aggregation

User Guide

FPGA-UG-02117 - 1.0

September 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

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
ACK	Acknowledge bit sent from RX side to TX side to indicate the received data parity check is OK
DAC	Digital to Analog Converter
FIFO	First In First Out
GPIO	General Purpose Inputs and Outputs
I2C	Inter-Integrated Circuit
I2S	Inter-IC Sound
PID	Payload ID
PLL	Phase Locked Loop
PT	Payload Type
RX	Receiver
TDM	Time Domain Multiplexing
TX	Transmitter
UART	Universal Asynchronous Receiver/Transmitter

1. General Description

This solution is an FPGA bitstream that can be used to configure a Lattice FPGA to perform as a Single-Wire aggregator for I^{2}C, I2S, UART*, and GPIO signalling. No FPGA design is needed. Two configured devices (denoted as Master Single-Wire Aggregation Device and Slave Single-Wire Aggregation Device) can be used to aggregate and deaggregate multiple signals on a single physical wire through Time Domain Multiplexing (TDM)-based bidirectional communication.

***Note:** UART can be implemented using GPIO channel.

1.1. Features

- Up to seven channels can be aggregated.
- Raw data rate on Single-Wire is configurable as ~7.5 Mbps or higher.
- Supports I^{2}C at 100 kpps, Fast-Mode (400 kbps) and Fast-Mode Plus (1 Mbps.)
- I^{2}C Interrupt can be realized by GPIO with event-based transmission.
- Supports up to 48 kHz sampling rate for I2S one-way stereo channel and 36 kHz sampling for two-way I2S stereo channel.[1]
- Supports ultra-low power devices.
  - As low as 130 µA standby current typical[2]
- Multiple sets of ready-to-use configurations to support different use case applications are available.
  - On-demand configuration request is possible through Lattice Technical Support.
- The configuration of the device is set through SRAM which provide the flexibility to be reprogram in the field.
  - SRAM is configured through:
  - Standard SPI Interface
  - Internal Non-volatile Configuration Memory (NVCM)
- Ultra-small form factor
  - 48 pin QFN (7 mm x 7 mm)
  - As small as 2.11 mm x 2.54 mm for 30 ball WLCSP[3]

Notes:

1. Configurations with I2S channel requires external clock as clock source.
2. For Configuration 1 at 25° C.
3. Ready-to-use configurations are only available at 48-pin QFN package. Minimal FPGA programming is required to support other package options.

1.2. Applications

- Client Computing: Display hinges, board to board connections.
- Consumer: Mobile device hinges, inter-module connections.
- Industrial and Automotive: Signal cables, sensor systems.

2. Block Diagram

Figure 2.1 shows a block diagram with I^{2}C, I2S, UART1, and GPIO aggregation, which has a Master Single-Wire Aggregation Device and a Slave Single-Wire Aggregation Device. There are three essential design differences between these devices:

- Upon power-up, the Master Single-Wire Aggregation Device generates a low pulse on the Single-Wire link and waits for another low pulse generated by the Slave Single-Wire Aggregation Device as an acknowledgement.
- For Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8 and Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx85, wherein I2S channels are available, the Master Single-Wire Aggregation device sends an I2S SCK pulse to the link for the Slave Single-Wire Aggregation device clock training. This is after the link acknowledgement by the Slave device.
- The Master Single-Wire Aggregation device always wins in the very first transmitter (TX) request contention after power-up.

***Note:** UART can be represented using GPIO channel.

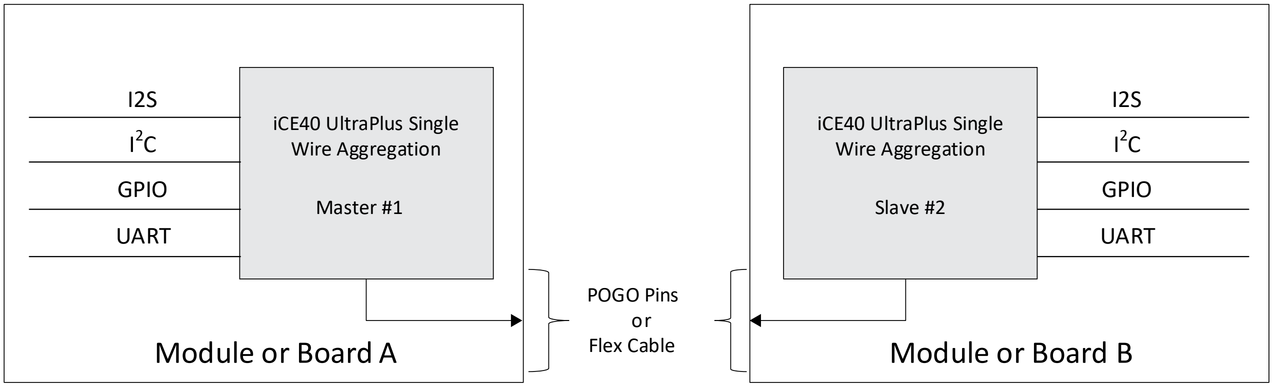


Figure 2.1. Single-Wire Aggregation Block Diagram

3. Ready-to-Use Device Configurations

3.1. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8

3.1.1. Supported Signals for Aggregation

- Two directional I2S channels (32 bits data width, 36 kHz audio sampling)
- One I^{2}C master to slave channel
- One I^{2}C slave to master channel
- 6 bits bidirectional GPIO channel
- 2 bits bidirectional GPIO channel

3.1.2. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8 Block Diagram

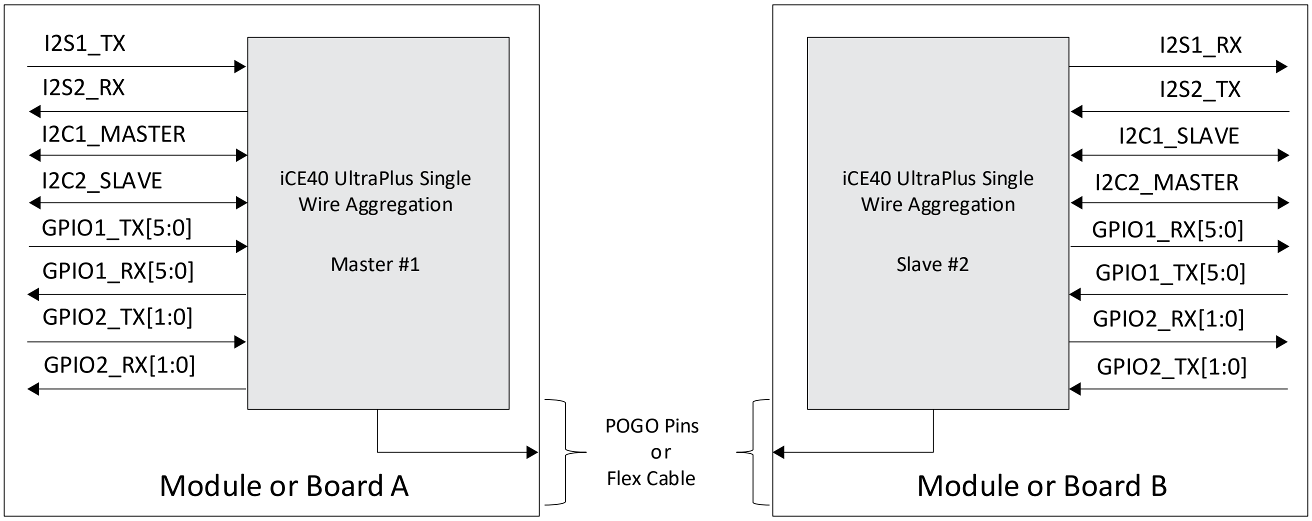


Figure 3.1. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8 Block Diagram

3.1.3. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8 Pin Information and Functions

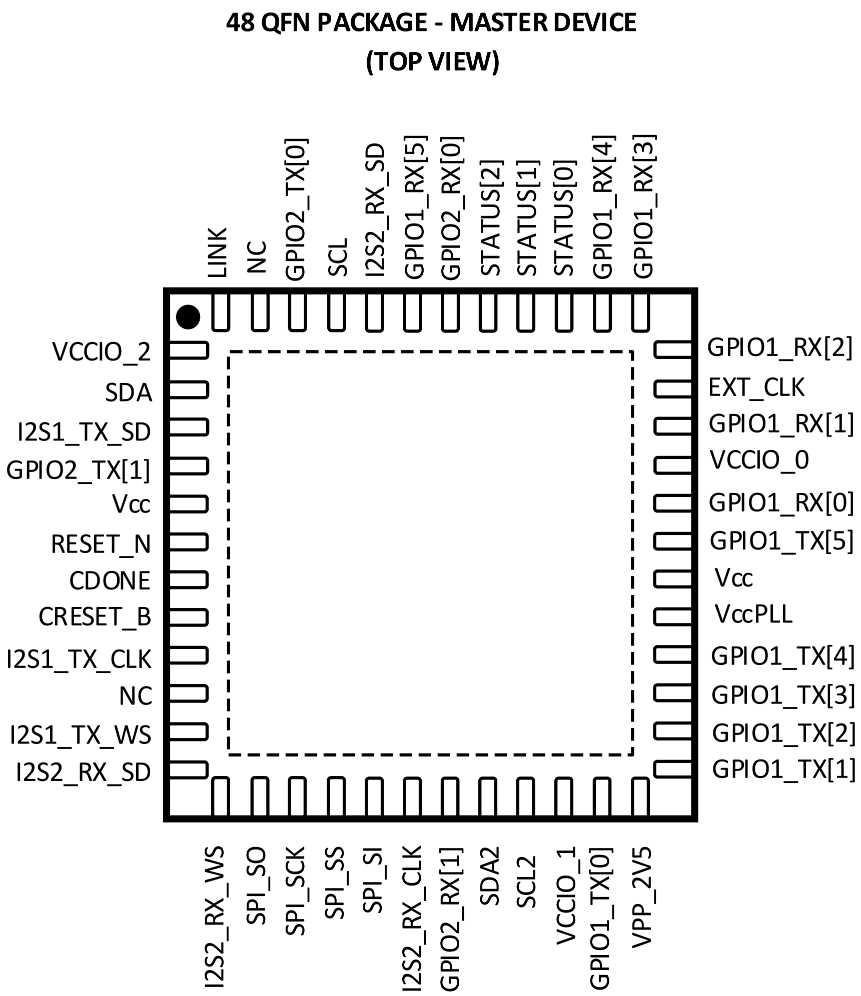


Figure 3.2. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8: Pin Configuration for Master Single-Wire Aggregation

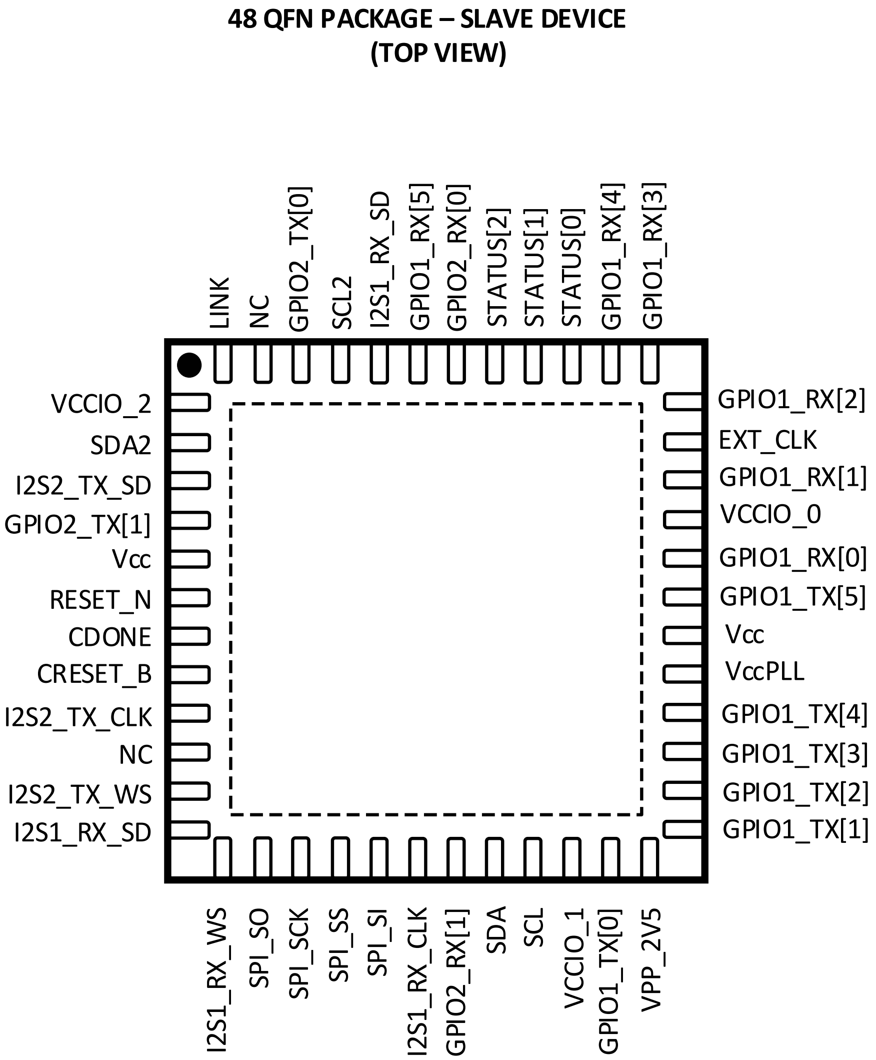


Figure 3.3. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8: Pin Configuration for Slave Single-Wire Aggregation

Table 3.1 provides the pin functions of Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8.

Table 3.1. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8 Pin Functions

PIN				TYPE	DESCRIPTION
NAME	I/O BANK	MASTER	SLAVE		
CDONE	1	7	7	CONFIG	Configuration Done. Includes a weak pull-up resistor to VCCIO_1.
CRESET_B	1	8	8	CONFIG	Configuration Reset, active LOW. No internal pull-up resistor. Either actively driven externally or connect a 10 kΩ pull-up to VCCIO_1.
EXT_CLK	0	35	35	I	12 MHz Input Clock to PLL
GND	—	PADDLE	PADDLE	GND	Ground
GPIO1_TX[0..5]	0	23,25,26,27,28,31	23,25,26,27,28,31	I	Channel 1 GPIO Transmitter
GPIO1_RX[0..5]	0	32,34,36,37,38,43	32,34,36,37,38,43	O	Channel 1 GPIO Receiver
GPIO2_TX[0,1]	1	46,4	46,4	I	Channel 2 GPIO Transmitter
GPIO2_RX[0,1]	2	42,19	42,19	O	Channel 2 GPIO Receiver
NC	-	10, 47	10, 47	—	Unused, with internal weak pull up
SCL	2	45	21	I/O	Channel 1 Serial Clock Line. With internal pull-up on VCCIO_1
SDA	2	2	20	I/O	Channel 1 Serial Data Line. With internal pull-up on VCCIO_1
SCL2	1	21	45	I/O	Channel 2 Serial Clock Line. With internal pull-up on VCCIO_1
SDA2	1	20	2	I/O	Channel 2 Serial Data Line. With internal pull-up on VCCIO_1
I2S1_RX_CLK	1		18	O	Channel 1 I2S_CLK Receiver
I2S1_RX_SD	1		12	O	Channel 1 I2S_SDA Receiver
I2S1_RX_WS	1		13	O	Channel 1 I2S_WS Receiver
I2S1_TX_CLK	1	9		O	Channel 1 I2S_CLK Transmitter (CONTROLLER), generate ~1 MHz clock signal to support 32 kHz Audio sampling
I2S1_TX_SD	2	3		O	Channel 1 I2S_SDA Transmitter
I2S1_TX_WS	1	11		I	Channel 1 I2S_WS Transmitter (CONTROLLER), generate I2S_WS ~32kHz clock signal to support 32kHz Audio sampling
I2S2_RX_CLK	1	18		O	Channel 2 I2S_CLK Receiver
I2S2_RX_SD	1	12		O	Channel 2 I2S_SDA Receiver
I2S2_RX_WS	1	13		O	Channel 2 I2S_WS Receiver
I2S2_TX_CLK	1		9	I	Channel 2 I2S_CLK Transmitter
I2S2_TX_SD	2		3	I	Channel 2 I2S_SDA Transmitter
I2S2_TX_WS	1		11	I	Channel 2 I2S_WS Transmitter
LINK	2	48	48	I/O	Single-Wire connection between Master and Slave Device. Requires external strong pull-up resistor.
RESET_N	1	6	6	I	System Reset, Active Low, with Internal Pull-up

PIN				TYPE	DESCRIPTION
NAME	I/O BANK	MASTER	SLAVE		
RGB0/STATUS[0]	0	39	39	LED	RGB LED Driver, Single-Wire Status[0]
RGB1/STATUS[1]	0	40	40	LED	RGB LED Driver, Single-Wire Status[1]
RGB2/STATUS[2]	0	41	41	LED	RGB LED Driver, Single-Wire Status[2]
SPI_SO	1	14	14	CONFIG_SPI	Configuration SPI SO
SPI_SS	1	16	16	CONFIG_SPI	Configuration SPI SS
SPI_SI	1	17	17	CONFIG_SPI	Configuration SPI SI
SPI_SCK	1	15	15	CONFIG_SPI	Configuration SPI SCK
Vcc	—	5,30	5,30	VCC	Core Power Supply
VCCIO_0	0	33	33	VCCIO	Power Supply for I/O Bank 0
VCCIO_1	1	22	22	VCCIO	Power Supply for I/O Bank 1
VCCIO_2	2	1	1	VCCIO	Power Supply for I/O Bank 2
VccPLL	—	29	29	VCCPLL	Power Supply for PLL
VPP_2V5	—	24	24	VPP	Power Supply for NVCM Programming and operations

3.1.4. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8 Typical Characteristics

T_A = 25°C (unless otherwise noted).

3.1.4.1. Master Single-Wire Aggregation Device

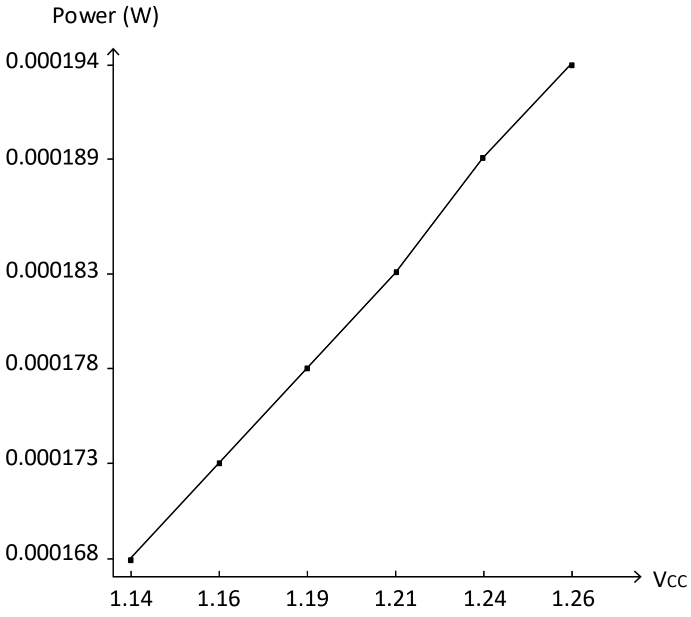


Figure 3.4. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8: Pin Configuration for Master Single-Wire Aggregation

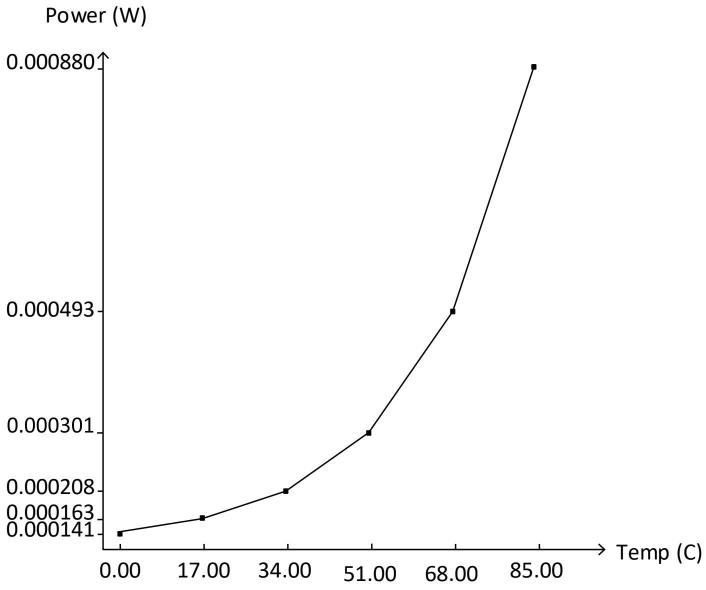


Figure 3.5. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8: Master Single-Wire Aggregation Device – Total Power versus Ambient Temperature (in °C)

3.1.4.2. Slave Single-Wire Aggregation Device

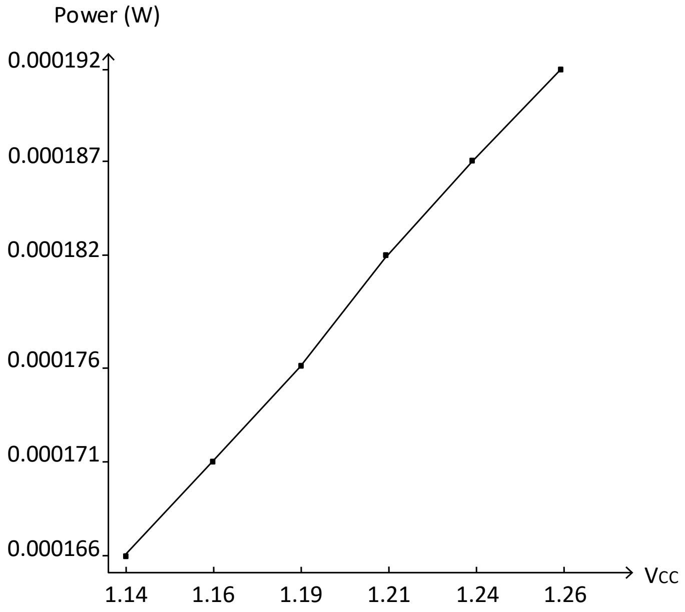


Figure 3.6. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8: Slave Single-Wire Aggregation Device – Total Power versus VCC (Volts)

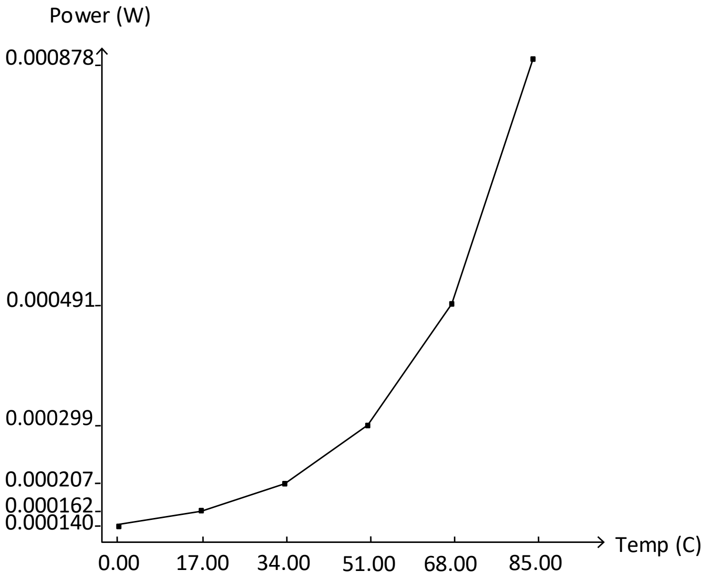


Figure 3.7. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8: Slave Single-Wire Aggregation Device – Total Power versus Ambient Temperature (in °C)

3.1.5. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8 Typical Propagation Delay

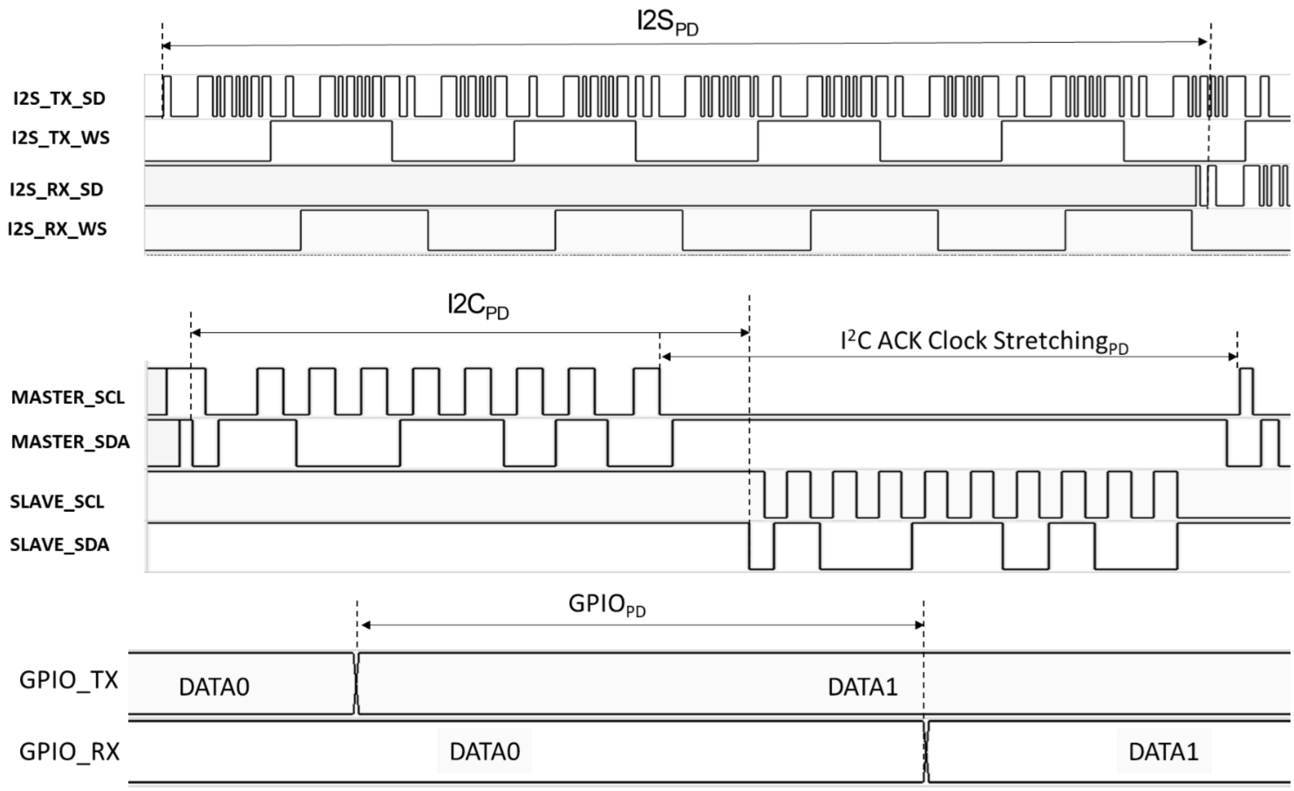


Figure 3.8. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8 Propagation Delay

Table 3.2. Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8 Typical Propagation Delay

	Propagation Delay (us)
$I^2C_{PD\ SCL}$ Clock at 400 kHz	26
I^2C ACK Clock Stretching $_{PD}$	24
$GPIO_{PD}$	3.8
$I2S_{PD}$	130

3.2. Configuration I2CMx6_GPIOx6

3.2.1. Supported Signals for Aggregation

- Six I^{2}C master to slave channels
- 6 bits master to slave GPIO channel

3.2.2. Configuration I2CMx6_GPIOx6 Block Diagram

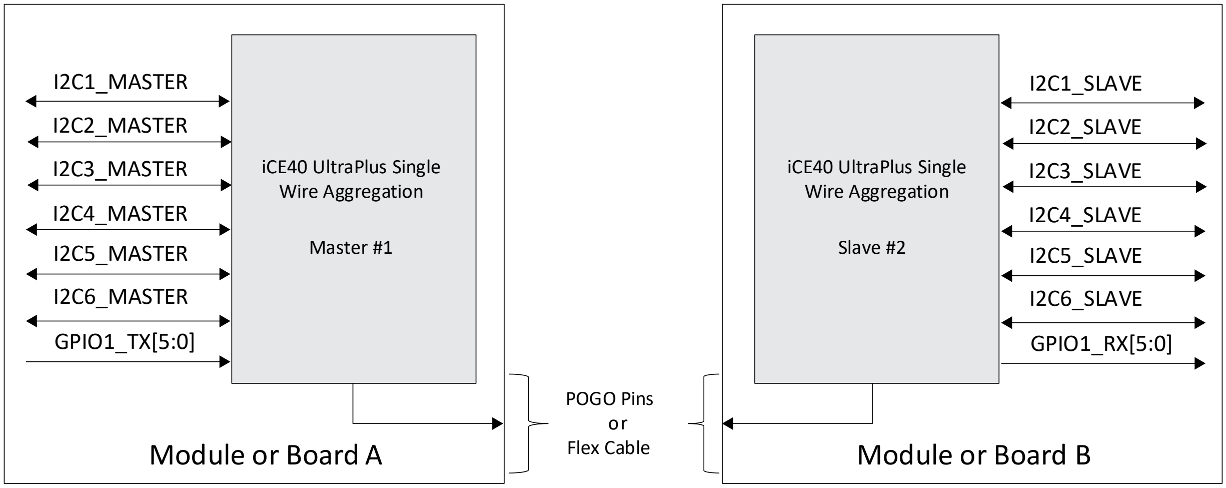


Figure 3.9. Configuration I2CMx6_GPIOx6 Block Diagram

3.2.3. Configuration I2CMx6_GPIOx6 Pin Information and Functions

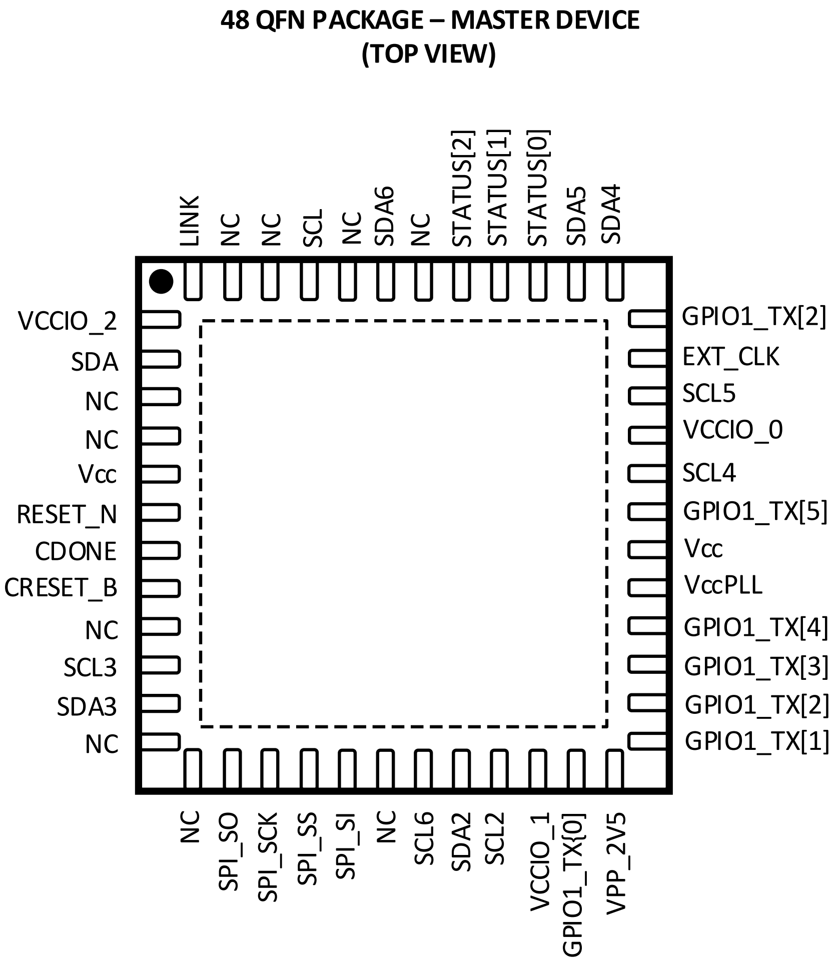


Figure 3.10. Configuration I2CMx6_GPIOx6: Pin Configuration for Master Single-Wire Aggregation

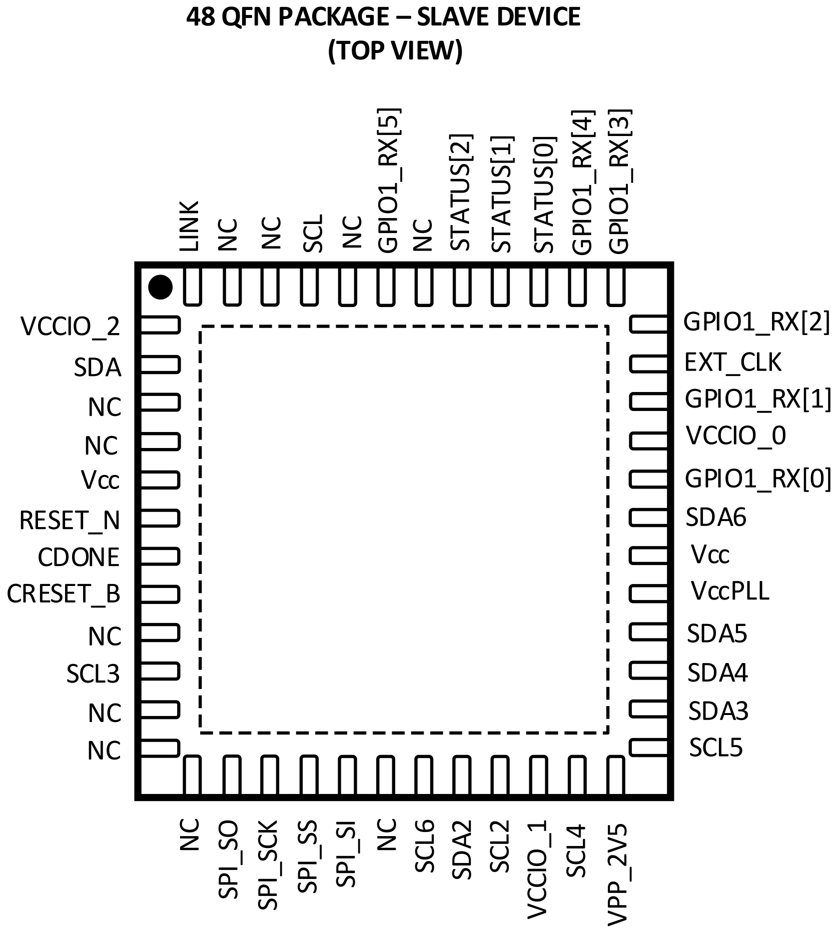


Figure 3.11. Configuration I2CMx6_GPIOx6: Pin Configuration for Slave Single-Wire Aggregation

Table 3.3 provides the pin functions of Configuration I2CMx6_GPIOx6.

Table 3.3. Configuration I2CMx6_GPIOx6 Pin Functions

PIN				TYPE	DESCRIPTION
NAME	I/O BANK	MASTER	SLAVE		
CDONE	1	7	7	CONFIG	Configuration Done. Includes a weak pull-up resistor to VCCIO_1.
CRESET_B	1	8	8	CONFIG	Configuration Reset, active LOW. No internal pull-up resistor. Either actively driven externally or connect a 10 kΩ pull-up to VCCIO_1.
EXT_CLK	0	35	35	I	12 MHz Input Clock to PLL
GND	—	PADDLE	PADDLE	GND	Ground
NC	—	3, 4, 9, 12, 13, 18, 42, 44, 46, 47	3, 4, 9, 11, 12, 13, 18, 42, 44, 46, 47	—	Unused, with internal weak pull up
GPIO1_TX[0,..5]	0	23, 25, 26, 27, 28, 31	—	I	Channel 1 GPIO Transmitter
GPIO1_RX[0,..5]	0	—	32, 34, 36, 37, 38, 43	O	Channel 1 GPIO Receiver
SCL	2	45	45	I/O	Channel 1 Serial Clock Line. With internal pull-up on VCCIO_1
SDA	2	2	2	I/O	Channel 1 Serial Data Line. With internal pull-up on VCCIO_1
SCL2	1	21	21	I/O	Channel 2 Serial Clock Line. With internal pull-up on VCCIO_1
SDA2	1	20	20	I/O	Channel 2 Serial Data Line. With internal pull-up on VCCIO_1
SCL3	1	10	10	I/O	Channel 3 Serial Clock Line. With internal pull-up on VCCIO_1
SDA3	1; 0	11	26	I/O	Channel 3 Serial Data Line. With internal pull-up on VCCIO_1
SCL4	0	32	23	I/O	Channel 4 Serial Clock Line. With internal pull-up on VCCIO_1
SDA4	0	37	37	I/O	Channel 4 Serial Data Line. With internal pull-up on VCCIO_1
SCL5	0	34	25	I/O	Channel 5 Serial Clock Line. With internal pull-up on VCCIO_0
SDA5	0	38	28	I/O	Channel 5 Serial Data Line. With internal pull-up on VCCIO_0
SCL6	0	19	19	I/O	Channel 6 Serial Clock Line. With internal pull-up on VCCIO_0
SDA6	0	43	31	I/O	Channel 6 Serial Data Line. With internal pull-up on VCCIO_0
LINK	2	48	48	I/O	Single-Wire connection between Master and Slave Device. Requires external strong pull-up resistor.
RESET_N	1	6	6	I	System Reset, Active Low, with Internal Pull-up
RGB0/STATUS[0]	0	39	39	LED	RGB LED Driver, Single-Wire Status[0]
RGB1/STATUS[1]	0	40	40	LED	RGB LED Driver, Single-Wire Status[1]
RGB2/STATUS[2]	0	41	41	LED	RGB LED Driver, Single-Wire Status[2]

PIN				TYPE	DESCRIPTION
NAME	I/O BANK	MASTER	SLAVE		
SPI_SO	1	14	14	CONFIG_SPI	Configuration SPI SO
SPI_SS	1	16	16	CONFIG_SPI	Configuration SPI SS
SPI_SI	1	17	17	CONFIG_SPI	Configuration SPI SI
SPI_SCK	1	15	15	CONFIG_SPI	Configuration SPI SCK
Vcc	—	5,30	5,30	VCC	Core Power Supply
VCCIO_0	0	33	33	VCCIO	Power Supply for I/O Bank 0
VCCIO_1	1	22	22	VCCIO	Power Supply for I/O Bank 1
VCCIO_2	2	1	1	VCCIO	Power Supply for I/O Bank 2
VccPLL	—	29	29	VCCPLL	Power Supply for PLL
VPP_2V5	—	24	24	VPP	Power Supply for NVCM Programming and operations

3.2.4. Configuration I2CMx6_GPIOx6 Typical Characteristics

T_A = 25°C (unless otherwise noted). Values are generated using Lattice Radiant® – Power Calculator tool.

3.2.4.1. Master Single-Wire Aggregation Device

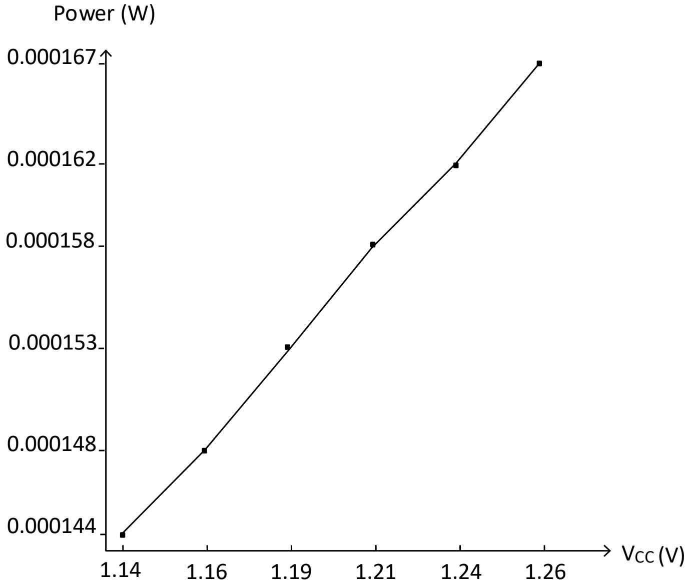


Figure 3.12. Configuration I2CMx6_GPIOx6: Master Single-Wire Aggregation Device – Total Power versus VCC (Volts)

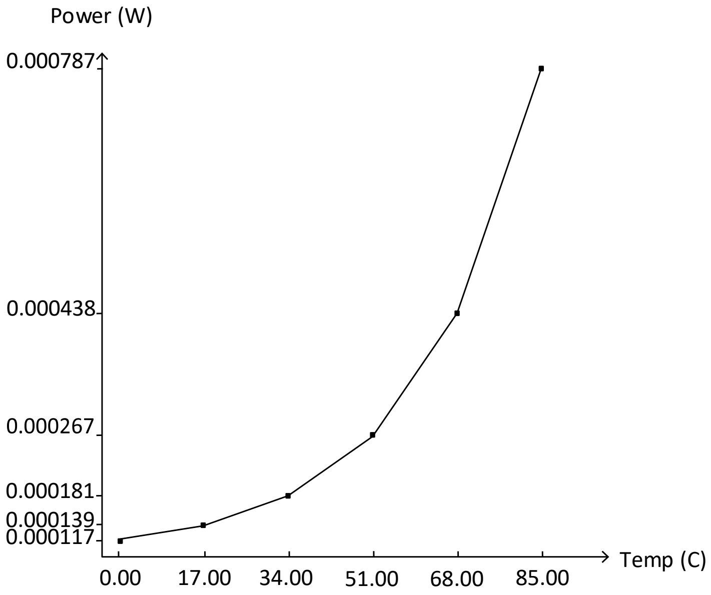


Figure 3.13. Configuration I2CMx6_GPIOx2: Master Single-Wire Aggregation Device – Total Power versus Ambient Temperature (in °C)

3.2.4.2. Slave Single-Wire Aggregation Device

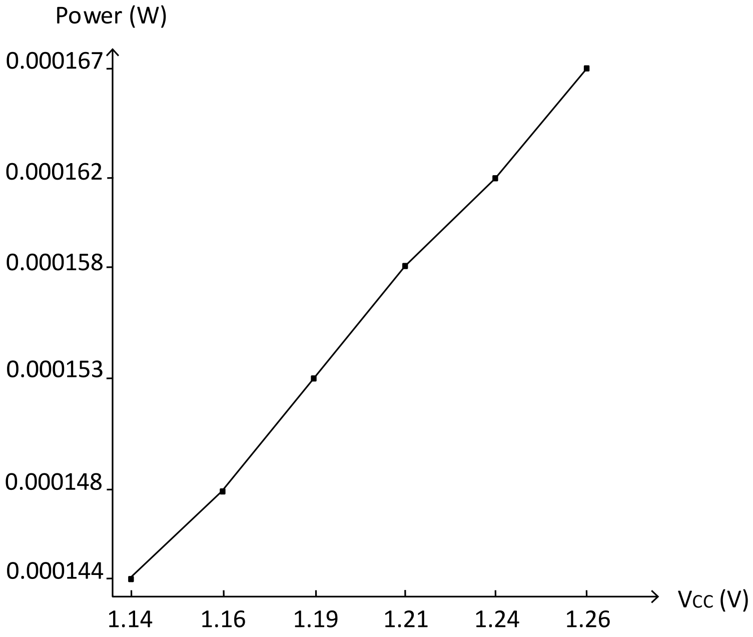


Figure 3.14. Configuration I2CMx6_GPIOx6 Slave Single-Wire Aggregation Device – Total Power versus VCC (Volts)

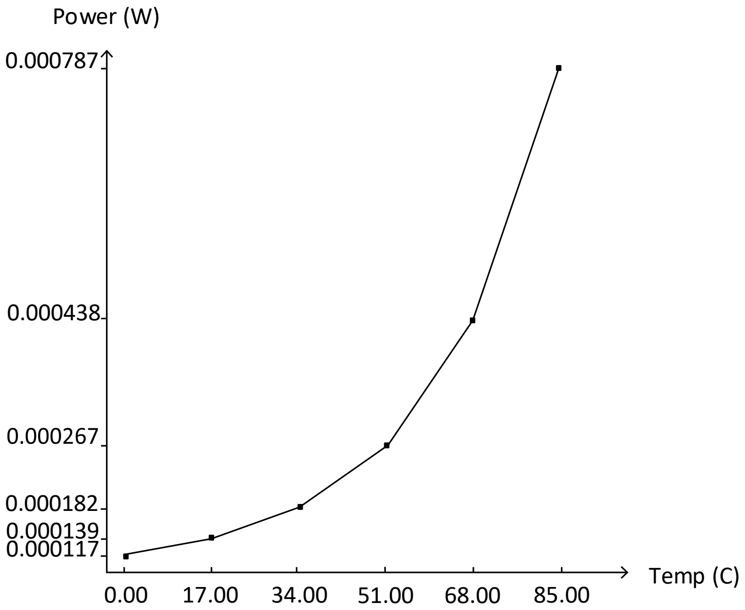


Figure 3.15. Configuration I2CMx6_GPIOx6: Slave Single-Wire Aggregation Device – Total Power versus Ambient Temperature (in °C)

3.2.5. Configuration I2CMx6_GPIOx6 Typical Propagation Delay

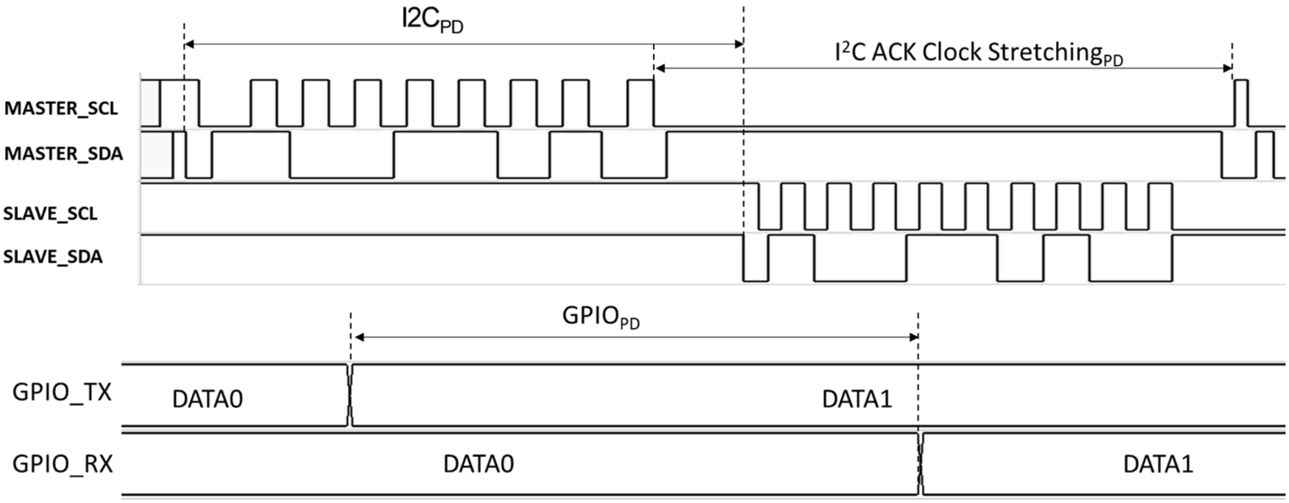


Figure 3.16. Configuration I2CMx6_GPIOx2: Propagation Delay

Table 3.4. Configuration I2CMx6_GPIOx6 Typical Propagation Delay

	Propagation Delay (us)
I^2C_{PD} (SCL clock at 400 kHz)	32
I^2C ACK Clock Stretching$_{PD}$	32
$GPIO_{PD}$	3.4

3.3. Configuration I2CMx1_GPIOx12

3.3.1. Supported Signals for Aggregation

- 1 MHz I^{2}C master to slave channel
- 12 bits bidirectional GPIO channel.

3.3.2. Configuration I2CMx1_GPIOx12 Block Diagram

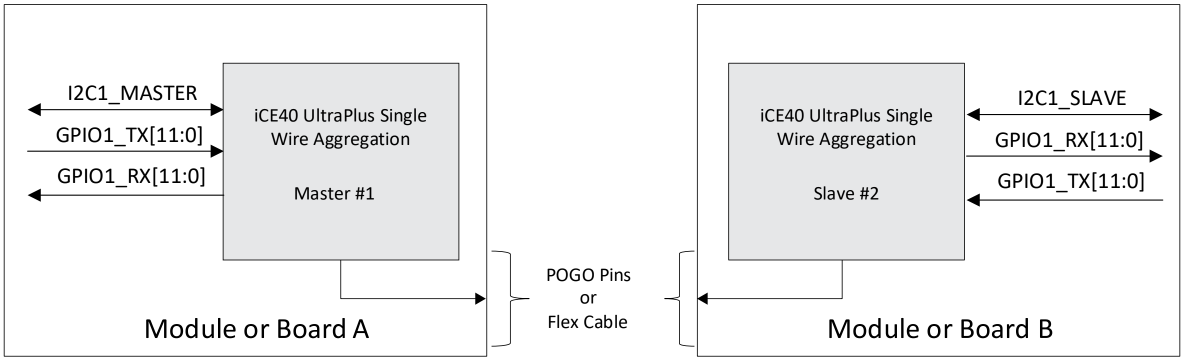


Figure 3.17. Configuration I2CMx1_GPIOx12 Block Diagram

3.3.3. Configuration I2CMx1_GPIOx12 Pin Information and Functions

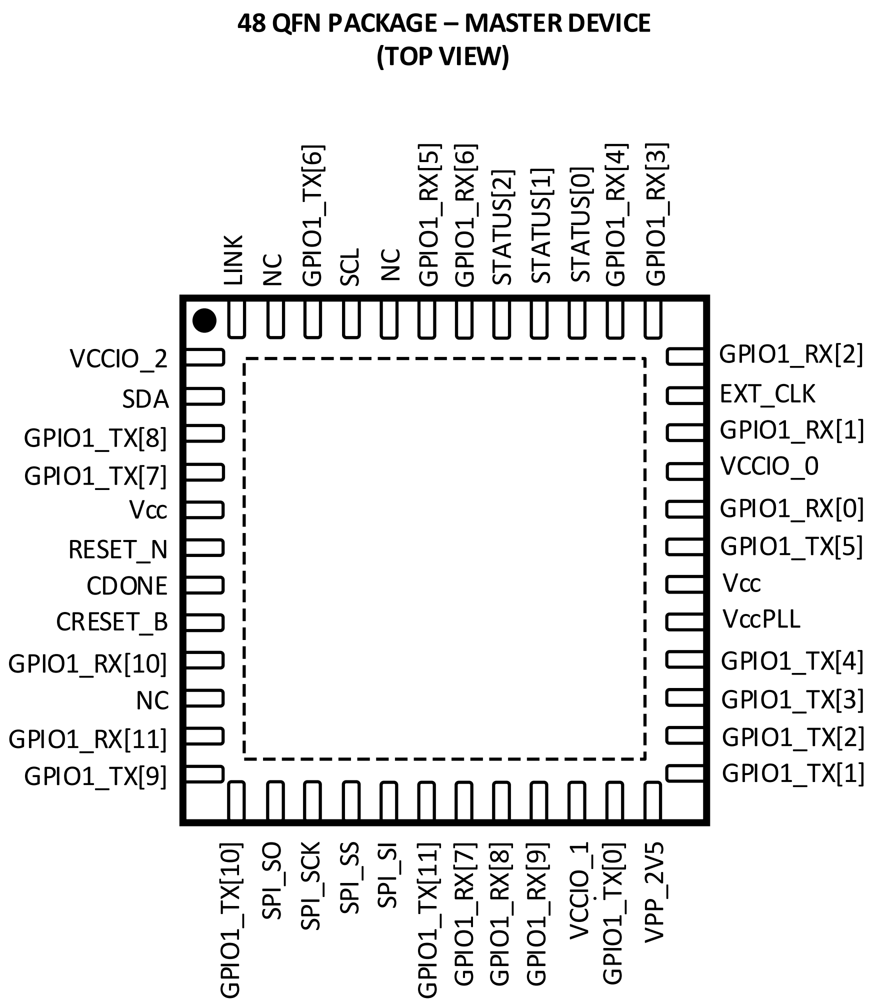


Figure 3.18. Configuration I2CMx1_GPIOx12: Pin Configuration for Master Single-Wire Aggregation

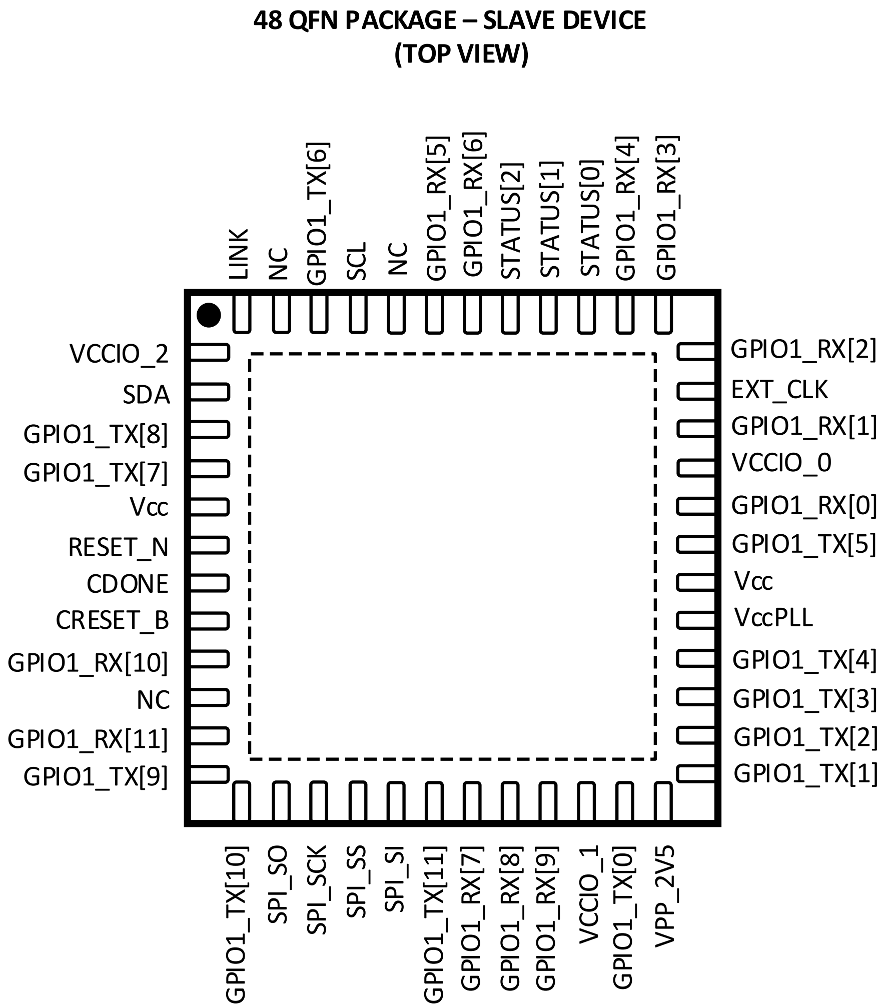


Figure 3.19. Configuration I2CMx1_GPIOx12: Pin Configuration for Slave Single-Wire Aggregation

Table 3.5 provides the pin functions of Configuration I2CMx1_GPIOx12.

Table 3.5. Configuration I2CMx1_GPIOx12 Pin Functions

PIN				TYPE	DESCRIPTION
NAME	I/O BANK	MASTER	SLAVE		
CDONE	1	7	7	CONFIG	Configuration Done. Includes a weak pull-up resistor to VCCIO_1.
CRESET_B	1	8	8	CONFIG	Configuration Reset, active LOW. No internal pull-up resistor. Either actively driven externally or connect a 10 kΩ pull-up to VCCIO_1.
EXT_CLK	0	35	35	I	12 MHz Input Clock to PLL
GND	—	PADDLE	PADDLE	GND	Ground
GPIO1_TX[0..11]	0,1,2	23, 25, 26, 27, 28, 31, 46, 4, 3, 12, 13, 18	23, 25, 26, 27, 28, 31, 46, 4, 3, 12, 13, 18	I	Channel 1 GPIO Transmitter
GPIO1_RX[0..11]	0,1	32, 34, 36, 37, 38, 43, 42, 19, 20, 21, 9, 11	32, 34, 36, 37, 38, 43, 42, 19, 20, 21, 9, 11	O	Channel 1 GPIO Receiver
NC	-	10, 44, 47	10, 44, 47	—	Unused, with internal weak pull up
SCL	2	45	45	I/O	Channel 1 Serial Clock Line. With internal pull-up on VCCIO_1
SDA	2	2	2	I/O	Channel 1 Serial Data Line. With internal pull-up on VCCIO_1
LINK	2	48	48	I/O	Single-Wire connection between Master and Slave Device. Requires external strong pull-up resistor.
RESET_N	1	6	6	I	System Reset, Active Low, with Internal Pull-up
RGB0/STATUS[0]	0	39	39	LED	RGB LED Driver, Single-Wire Status[0]
RGB1/STATUS[1]	0	40	40	LED	RGB LED Driver, Single-Wire Status[1]
RGB2/STATUS[2]	0	41	41	LED	RGB LED Driver, Single-Wire Status[2]
SPI_SO	1	14	14	CONFIG_SPI	Configuration SPI SO
SPI_SS	1	16	16	CONFIG_SPI	Configuration SPI SS
SPI_SI	1	17	17	CONFIG_SPI	Configuration SPI SI
SPI_SCK	1	15	15	CONFIG_SPI	Configuration SPI SCK
Vcc	—	5,30	5,30	VCC	Core Power Supply
VCCIO_0	0	33	33	VCCIO	Power Supply for IO Bank 0
VCCIO_1	1	22	22	VCCIO	Power Supply for IO Bank 1
VCCIO_2	2	1	1	VCCIO	Power Supply for IO Bank 2
VccPLL	—	29	29	VCCPLL	Power Supply for PLL
VPP_2V5	—	24	24	VPP	Power Supply for NVCM Programming and operations

3.3.4. Configuration I2CMx1_GPIOx12 Typical Characteristics

T_A = 25°C (unless otherwise noted). Values are generated using Lattice Radiant – Power Calculator tool.

3.3.4.1. Master Single-Wire Aggregation Device

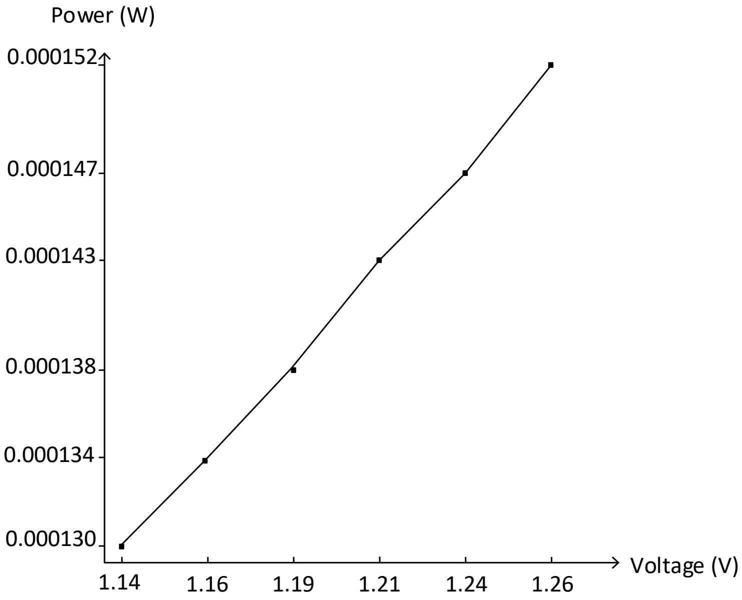


Figure 3.20. Configuration I2CMx1_GPIOx12: Master Single-Wire Aggregation Device – Total Power versus VCC (Volts)

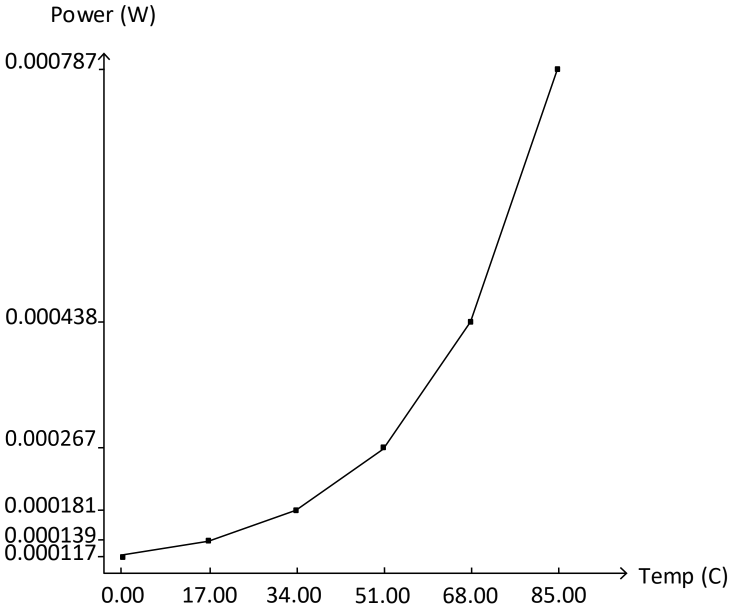


Figure 3.21. Configuration I2CMx1_GPIOx12: Master Single-Wire Aggregation Device – Total Power versus Ambient Temperature (in °C)

3.3.4.2. Slave Single-Wire Aggregation Device

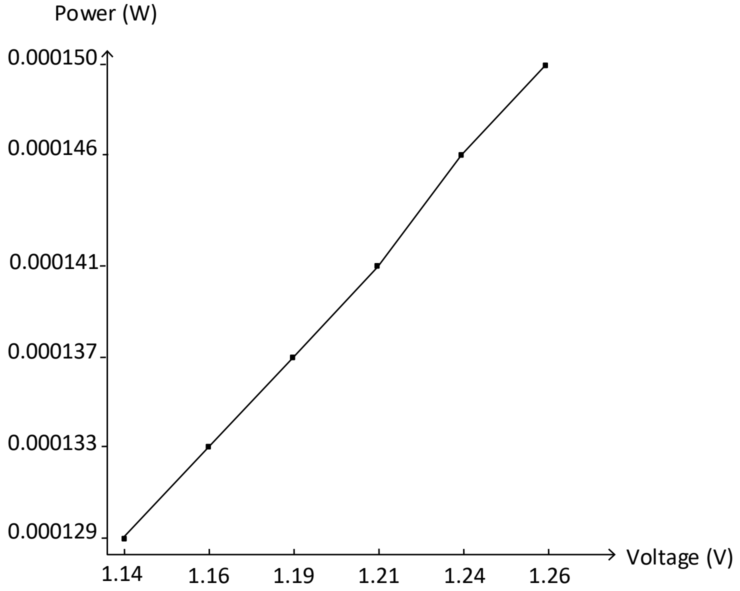


Figure 3.22. Configuration I2CMx1_GPIOx12: Slave Single-Wire Aggregation Device – Total Power versus VCC (Volts)

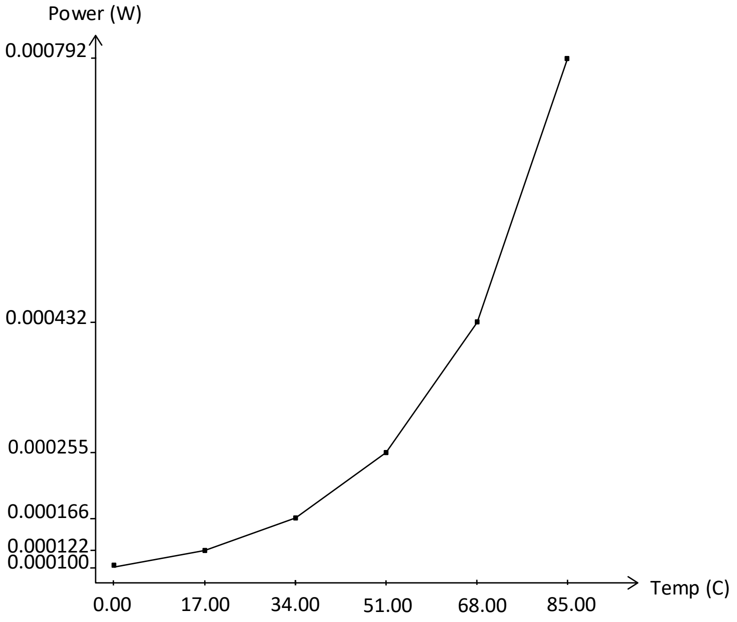


Figure 3.23. Configuration I2CMx1_GPIOx12: Slave Single-Wire Aggregation Device – Total Power versus Ambient Temperature (in °C)

3.3.5. Configuration I2CMx1_GPIOx12 Typical Propagation Delay

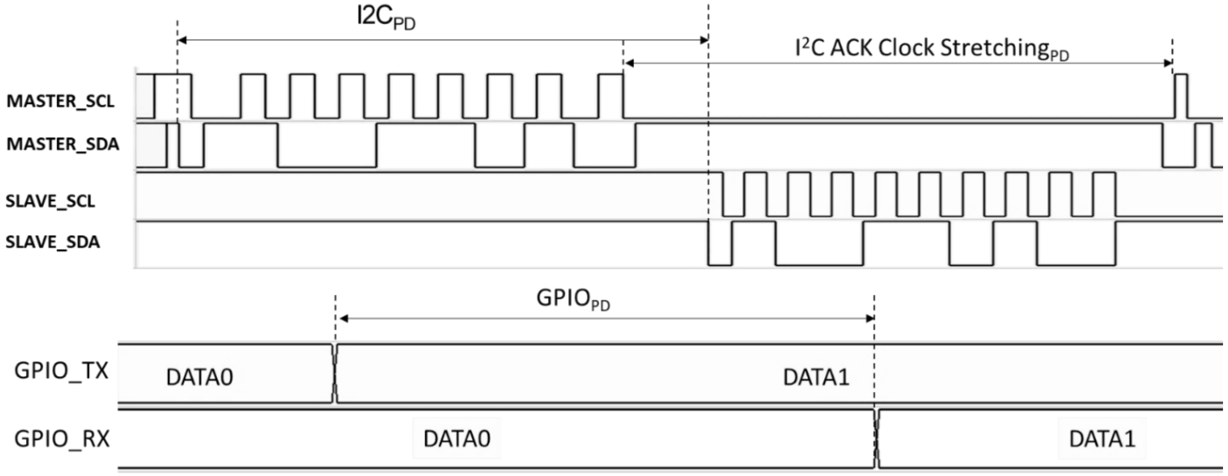


Figure 3.24. Configuration I2CMx1_GPIOx12 Propagation Delay

Table 3.6. Configuration I2CMx1_GPIOx12 Typical Propagation Delay

	Propagation Delay (us)
I^2C_{PD} (SCL Clock at 1 MHz)	13
I^2C ACK Clock $Stretching_{PD}$	13
$GPIO_{PD}$	1.9

3.4. Configuration I2CMx3_I2CSx2_GPIOx15

3.4.1. Supported Signals for Aggregation

- Three I²C master to slave channels
- Two I²C slave to master channels
- 15 bits master to slave GPIO channel

3.4.2. Configuration I2CMx3_I2CSx2_GPIOx15 Block Diagram

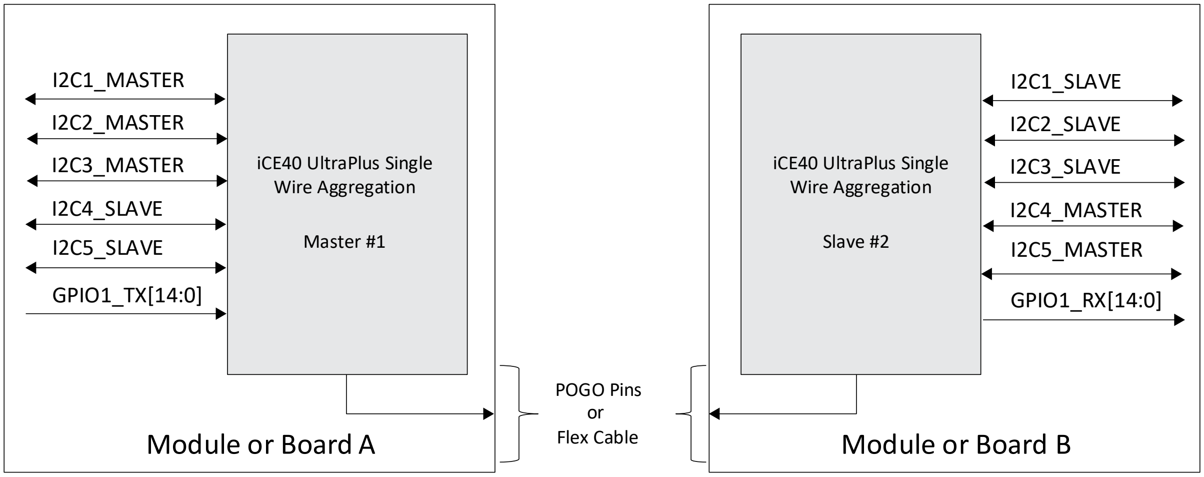


Figure 3.25. Configuration I2CMx3_I2CSx2_GPIOx15 Block Diagram

3.4.3. Configuration I2CMx3_I2CSx2_GPIOx15 Pin Information and Functions

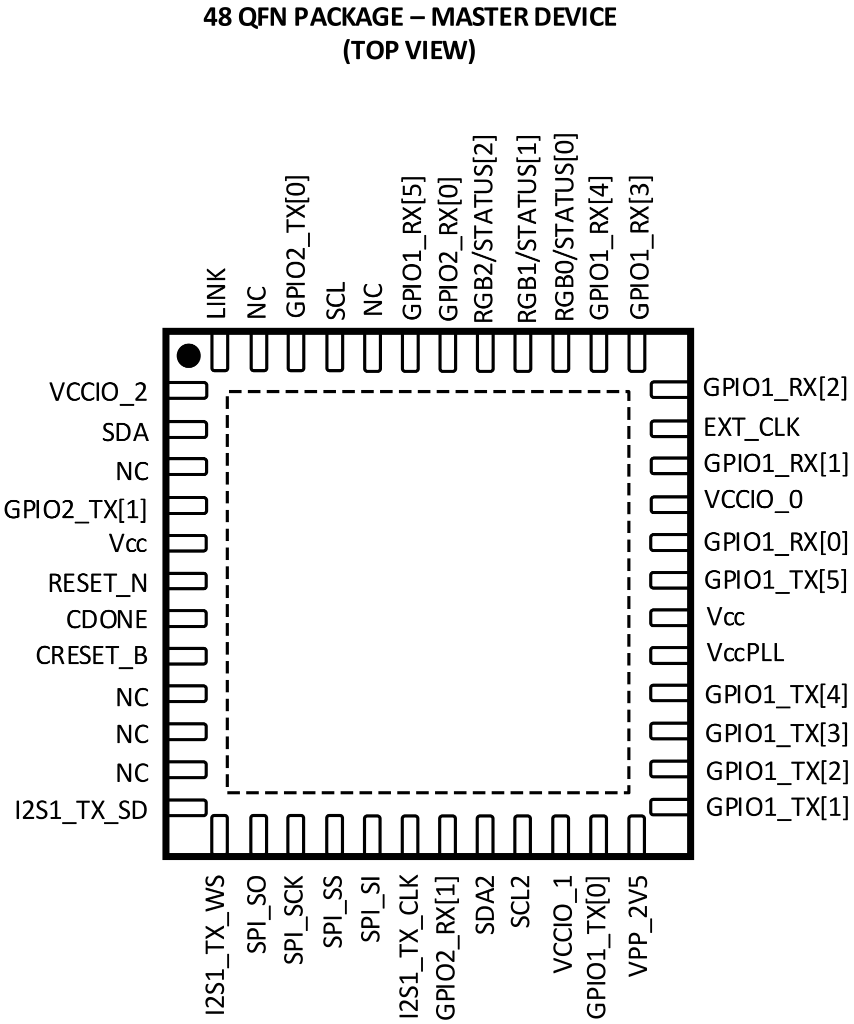


Figure 3.26. Configuration I2CMx3_I2CSx2_GPIOx15: Pin Configuration for Master Single-Wire Aggregation

**48 QFN PACKAGE – SLAVE DEVICE
(TOP VIEW)**

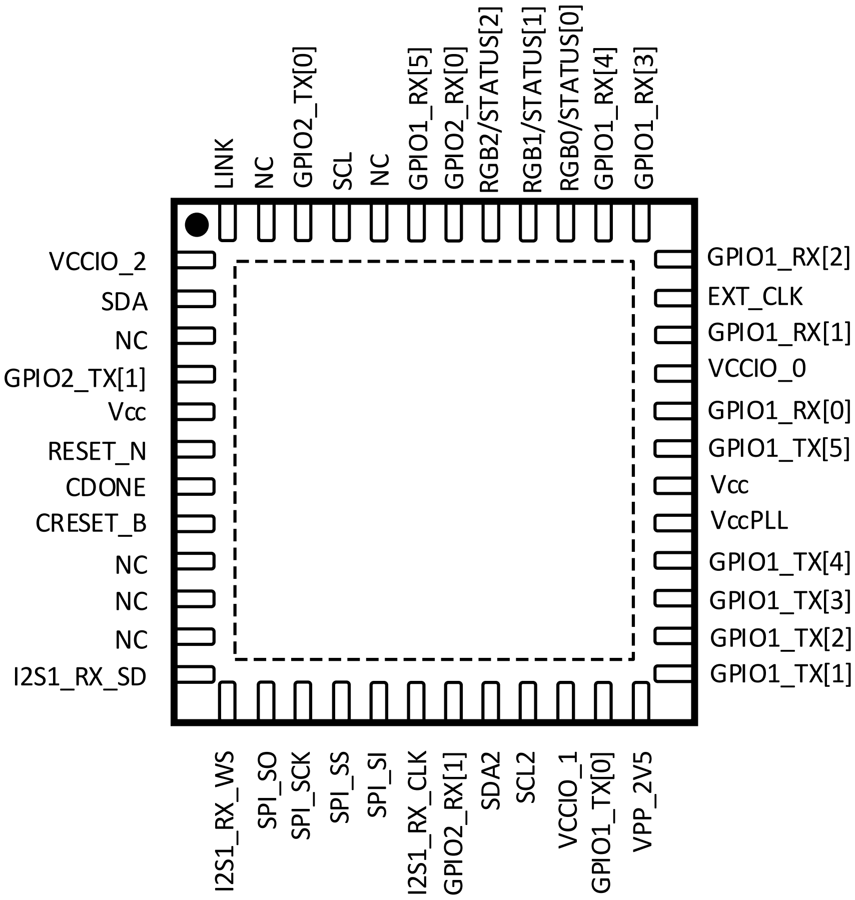


Figure 3.27. Configuration I2CMx3_I2CSx2_GPIOx15: Pin Configuration for Slave Single-Wire Aggregation

Table 3.7 provides the pin functions of Configuration I2CMx3_I2CSx2_GPIOx15.

Table 3.7. Configuration I2CMx3_I2CSx2_GPIOx15 Pin Functions

PIN				TYPE	DESCRIPTION
NAME	I/O BANK	MASTER	SLAVE		
CDONE	1	7	7	CONFIG	Configuration Done. Includes a weak pull-up resistor to VCCIO_1.
CRESET_B	1	8	8	CONFIG	Configuration Reset, active LOW. No internal pull-up resistor. Either actively driven externally or connect a 10 kΩ pull-up to VCCIO_1.
EXT_CLK	0	35	35	I	12 MHz Input Clock to PLL
GND	—	PADDLE	PADDLE	GND	Ground
NC	—	10,18,22,44,47	10,18,22,44,47	—	Unused, with internal weak pull up
GPIO1_TX[0,14]	0,1	24,25,26,27,28,31,46,32,34,36,37,38,43,42,19	—	I	Channel 1 GPIO Transmitter
GPIO1_RX[0,14]	0,1	—	32,34,36,37,38,43,42,19,23,25,26,27,28,31	O	Channel 1 GPIO Receiver
SCL	2	45	45	IO	Channel 1 Serial Clock Line. With internal pull-up on VCCIO_1
SDA	2	2	2	IO	Channel 1 Serial Data Line. With internal pull-up on VCCIO_1
SCL2	1	21	21	IO	Channel 2 Serial Clock Line. With internal pull-up on VCCIO_1
SDA2	1	20	20	IO	Channel 2 Serial Data Line. With internal pull-up on VCCIO_1
SCL3	2	4	4	IO	Channel 3 Serial Clock Line. With internal pull-up on VCCIO_1
SDA3	2	3	3	IO	Channel 3 Serial Data Line. With internal pull-up on VCCIO_1
SCL4	1	11	11	IO	Channel 4 Serial Clock Line. With internal pull-up on VCCIO_1
SDA4	1	12	12	IO	Channel 4 Serial Data Line. With internal pull-up on VCCIO_1
SCL5	1	13	14	IO	Channel 5 Serial Clock Line. With internal pull-up on VCCIO_0
SDA5	1	9	9	IO	Channel 5 Serial Data Line. With internal pull-up on VCCIO_0
LINK	2	48	48	IO	Single-Wire connection between Master and Slave Device. Requires external strong pull-up resistor.
RESET_N	1	6	6	I	System Reset, Active Low, with Internal Pull-up
RGB0/STATUS[0]	0	39	39	LED	RGB LED Driver, Single-Wire Status[0]
RGB1/STATUS[1]	0	40	40	LED	RGB LED Driver, Single-Wire Status[1]
RGB2/STATUS[2]	0	41	41	LED	RGB LED Driver, Single-Wire Status[2]

PIN				TYPE	DESCRIPTION
NAME	I/O BANK	MASTER	SLAVE		
SPI_SO	1	14	14	CONFIG_SPI	Configuration SPI SO
SPI_SS	1	16	16	CONFIG_SPI	Configuration SPI SS
SPI_SI	1	17	17	CONFIG_SPI	Configuration SPI SI
SPI_SCK	1	15	15	CONFIG_SPI	Configuration SPI SCK
Vcc	—	5,30	5,30	VCC	Core Power Supply
VCCIO_0	0	33	33	VCCIO	Power Supply for IO Bank 0
VCCIO_1	1	22	22	VCCIO	Power Supply for IO Bank 1
VCCIO_2	2	1	1	VCCIO	Power Supply for IO Bank 2
VccPLL	—	29	29	VCCPLL	Power Supply for PLL
VPP_2V5	—	24	24	VPP	Power Supply for NVCM Programming and operations

3.4.4. Configuration I2CMx3_I2CSx2_GPIOx15 Typical Characteristics

T_A = 25°C (unless otherwise noted). Values are generated using Lattice Radiant – Power Calculator tool).

3.4.4.1. Master Single-Wire Aggregation Device

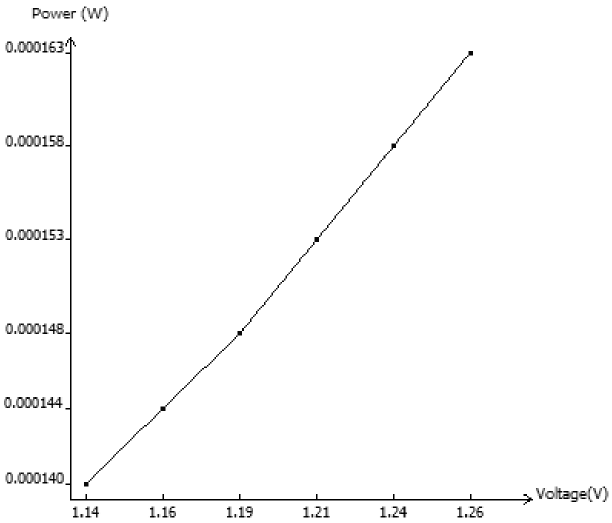


Figure 3.28. Configuration I2CMx3_I2CSx2_GPIOx15: Total Power versus VCC (Volts)

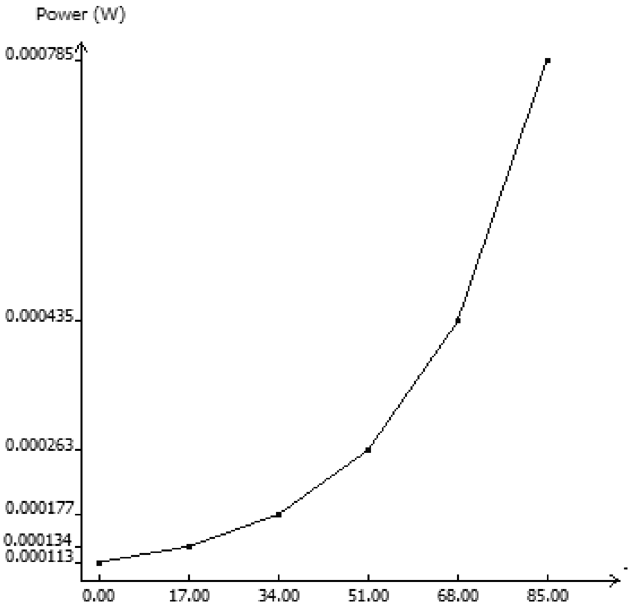


Figure 3.29. Configuration I2CMx3_I2CSx2_GPIOx15: Master Single-Wire Aggregation Device – Total Power versus Ambient Temperature (in °C)

3.4.4.2. Slave Single-Wire Aggregation Device

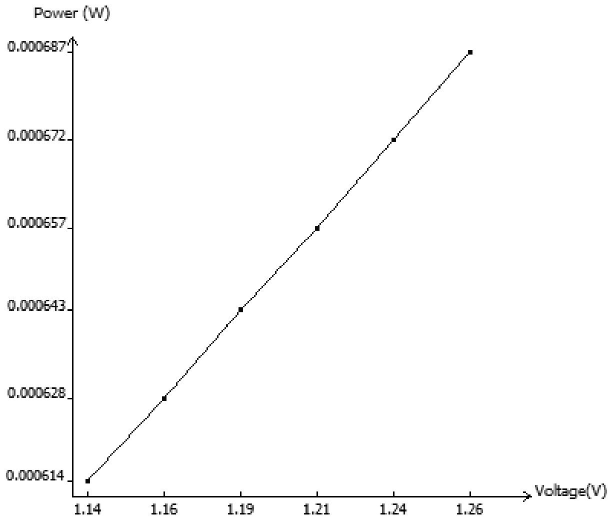


Figure 3.30. Configuration I2CMx3_I2CSx2_GPIOx15: Slave Single-Wire Aggregation Device – Total Power versus VCC (Volts)

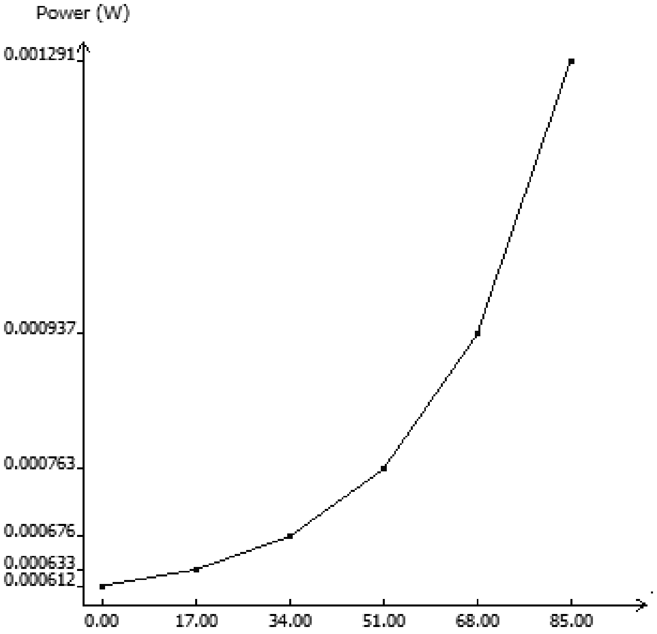


Figure 3.31. Configuration I2CMx3_I2CSx2_GPIOx15: Slave Single-Wire Aggregation Device – Total Power versus Ambient Temperature (in °C)

3.4.5. Configuration I2CMx3_I2CSx2_GPIOx15 Typical Propagation Delay

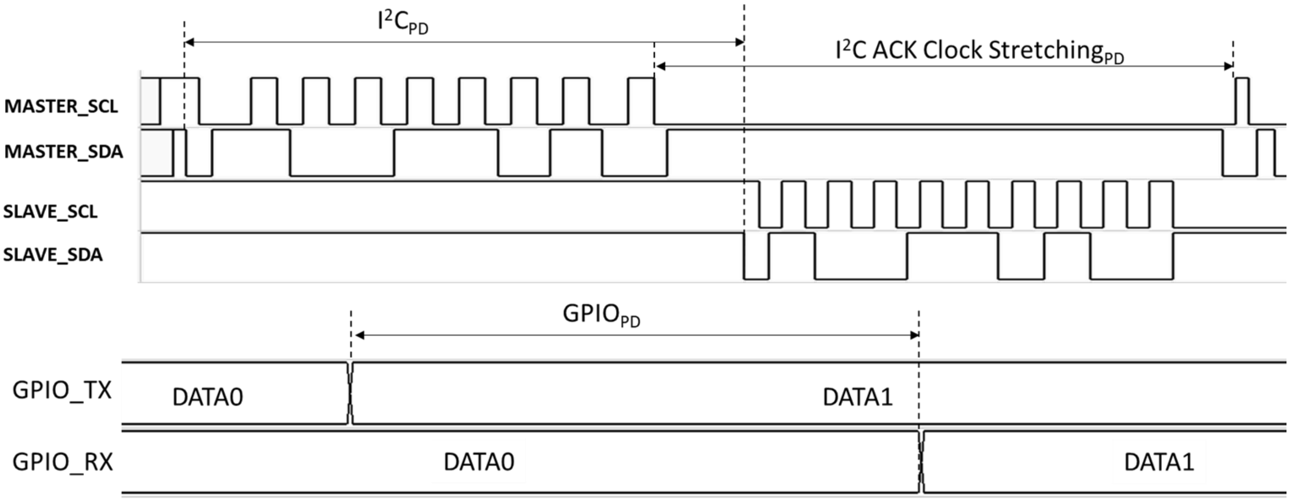


Figure 3.32. Configuration I2CMx3_I2CSx2_GPIOx15 Propagation Delay

Table 3.8. Configuration I2CMx3_I2CSx2_GPIOx15 Typical Propagation Delay

	Propagation Delay (us)
I^2C_{PD} (SCL Clock at 400 kHz)	33.3
I^2C ACK Clock Stretching $_{PD}$	28.7
$GPIO_{PD}$	2.1

3.5. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8

3.5.1. Supported Signals for Aggregation

- One directional I2S channels (32 bits data width, up to 48 kHz audio sampling)
- One I^2C master to slave channel
- One I^2C slave to master channel
- 6 bits bidirectional GPIO channel
- 2 bits bidirectional GPIO channel

3.5.2. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8 Block Diagram

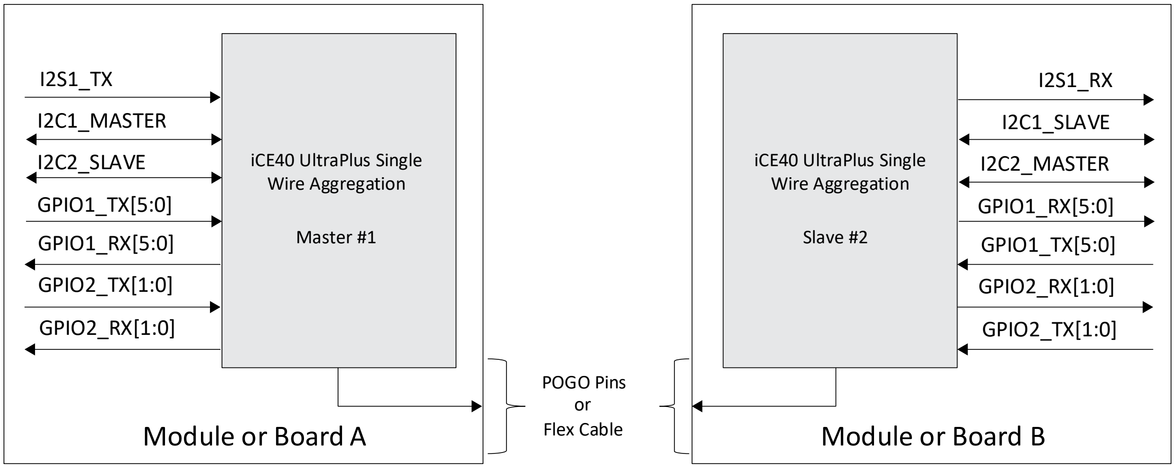


Figure 3.33. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8 Block Diagram

3.5.3. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8 Pin Information and Functions

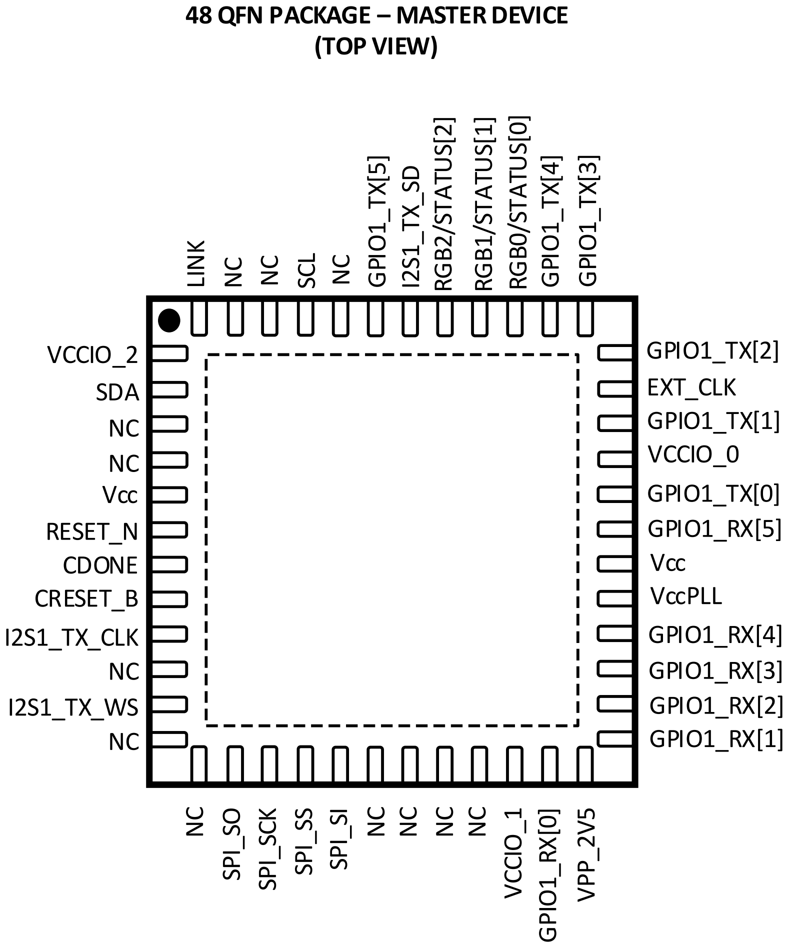


Figure 3.34. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8: Pin Configuration for Master Single-Wire Aggregation

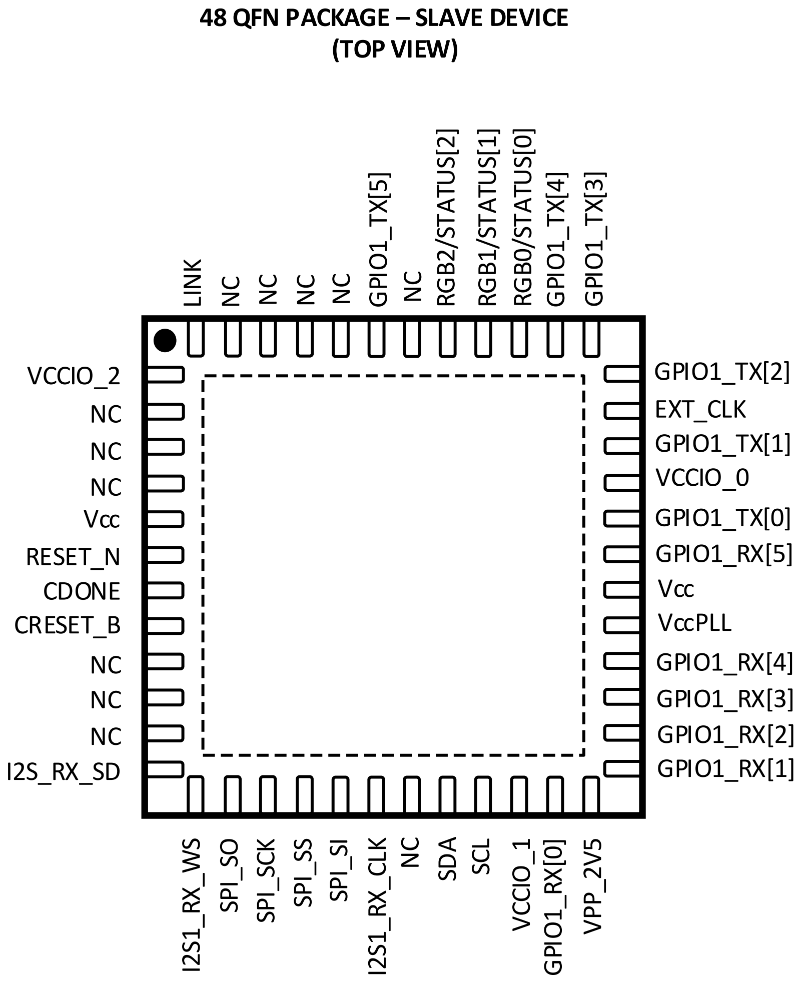


Figure 3.35. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8: Pin Configuration for Slave Single-Wire Aggregation

Table 3.9 provides the pin functions of Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8.

Table 3.9. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8 Pin Functions

PIN				TYPE	DESCRIPTION
NAME	I/O BANK	MASTER	SLAVE		
CDONE	1	7	7	CONFIG	Configuration Done. Includes a weak pull-up resistor to VCCIO_1.
CRESET_B	1	8	8	CONFIG	Configuration Reset, active LOW. No internal pull-up resistor. Either actively driven externally or connect a 10 kΩ pull-up to VCCIO_1.
EXT_CLK	0	35	35	I	12 MHz Input Clock to PLL
GND	—	PADDLE	PADDLE	GND	Ground
GPIO1_TX[0..5]	0	32,34,36,37,38,42	32,34,36,37,38,42	O	Channel 1 GPIO Transmitter
GPIO1_RX[0..5]	0	23,25,26,27,28,31	23,25,26,27,28,31	I	Channel 1 GPIO Receiver
GPIO2_TX[0,1]	0,1	42,19	42,19	O	Channel 2 GPIO Transmitter
GPIO2_RX[0,1]	2	46,4	46,4	I	Channel 2 GPIO Receiver

PIN				TYPE	DESCRIPTION
NAME	I/O BANK	MASTER	SLAVE		
NC	—	—	—	—	Unused, with internal weak pull up
SCL	2	45	45	I/O	Channel 1 Serial Clock Line. With internal pull-up on VCCIO_1
SDA	2	2	2	I/O	Channel 1 Serial Data Line. With internal pull-up on VCCIO_1
SCL2	1	21	21	I/O	Channel 2 Serial Clock Line. With internal pull-up on VCCIO_1
SDA2	1	20	20	I/O	Channel 2 Serial Data Line. With internal pull-up on VCCIO_1
I2S1_RX_CLK	1		18	O	Channel 1 I2S_CLK Receiver
I2S1_RX_SD	1		12	O	Channel 1 I2S_SDA Receiver
I2S1_RX_WS	1		13	O	Channel 1 I2S_WS Receiver
I2S1_TX_CLK	1	9	—	I	Channel 1 I2S_CLK Transmitter
I2S1_TX_SD	2	3	—	I	Channel 1 I2S_SDA Transmitter
I2S1_TX_WS	1	11	—	I	Channel 1 I2S_WS Transmitter
LINK	2	48	48	I/O	Single-Wire connection between Master and Slave Device. Requires external strong pull-up resistor.
RESET_N	1	6	6	I	System Reset, Active Low, with Internal Pull-up
RGB0/STATUS[0]	0	39	39	LED	RGB LED Driver, Single-Wire Status[0]
RGB1/STATUS[1]	0	40	40	LED	RGB LED Driver, Single-Wire Status[1]
RGB2/STATUS[2]	0	41	41	LED	RGB LED Driver, Single-Wire Status[2]
SPI_SO	1	14	14	CONFIG_SPI	Configuration SPI SO
SPI_SS	1	16	16	CONFIG_SPI	Configuration SPI SS
SPI_SI	1	17	17	CONFIG_SPI	Configuration SPI SI
SPI_SCK	1	15	15	CONFIG_SPI	Configuration SPI SCK
Vcc	—	5,30	5,30	VCC	Core Power Supply
VCCIO_0	0	33	33	VCCIO	Power Supply for I/O Bank 0
VCCIO_1	1	22	22	VCCIO	Power Supply for I/O Bank 1
VCCIO_2	2	1	1	VCCIO	Power Supply for I/O Bank 2
VccPLL	—	29	29	VCCPLL	Power Supply for PLL
VPP_2V5	—	24	24	VPP	Power Supply for NVCM Programming and operations

3.5.4. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8 Typical Characteristics

T_A = 25°C (unless otherwise noted). Values are generated using Lattice Radiant – Power Calculator tool.

3.5.4.1. Master Single-Wire Aggregation Device

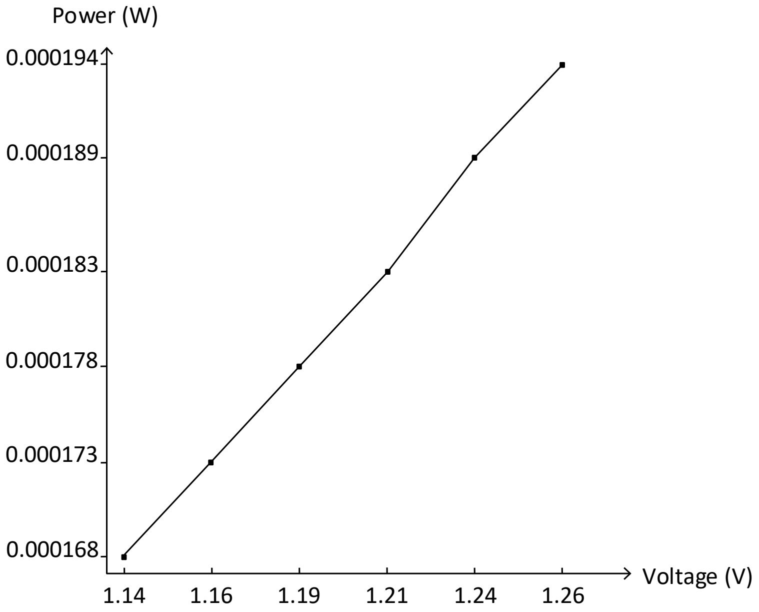


Figure 3.36. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8: Master Single-Wire Aggregation Device – Total Power versus VCC (Volts)

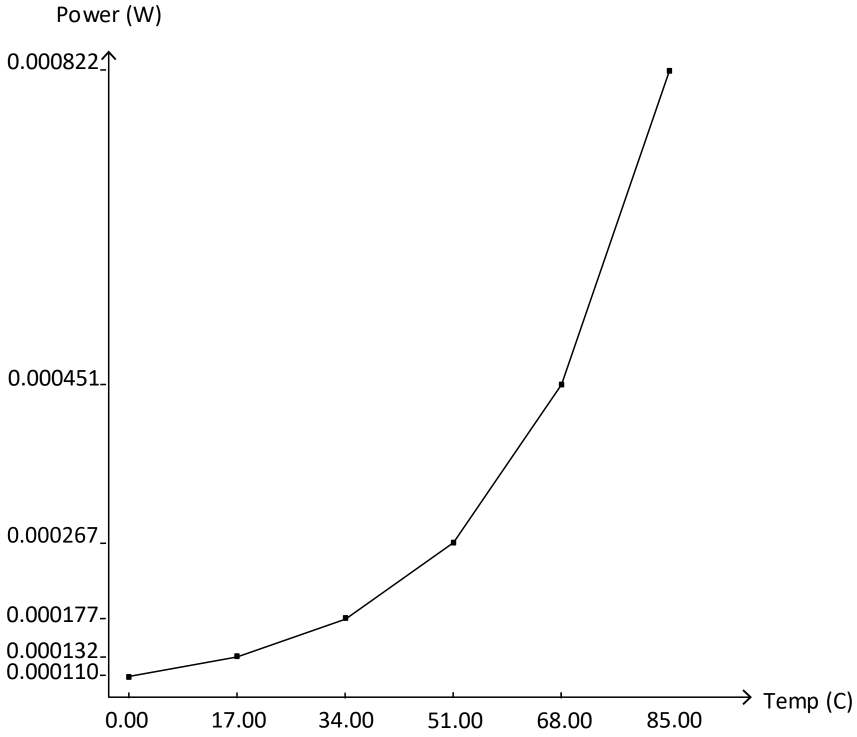


Figure 3.37. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8: Master Single-Wire Aggregation Device – Total Power versus Ambient Temperature (in °C)

3.5.4.2. Slave Single-Wire Aggregation Device

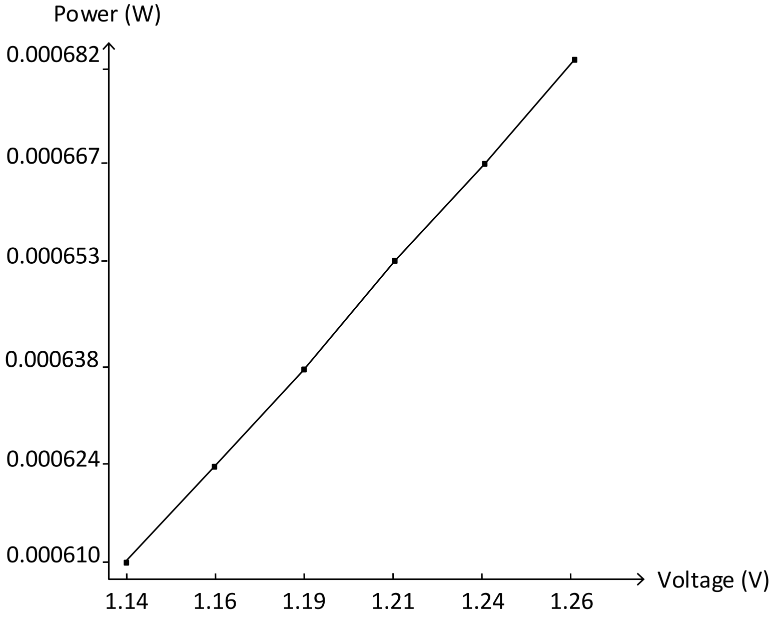


Figure 3.38. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8: Slave Single-Wire Aggregation Device – Total Power versus VCC (Volts)

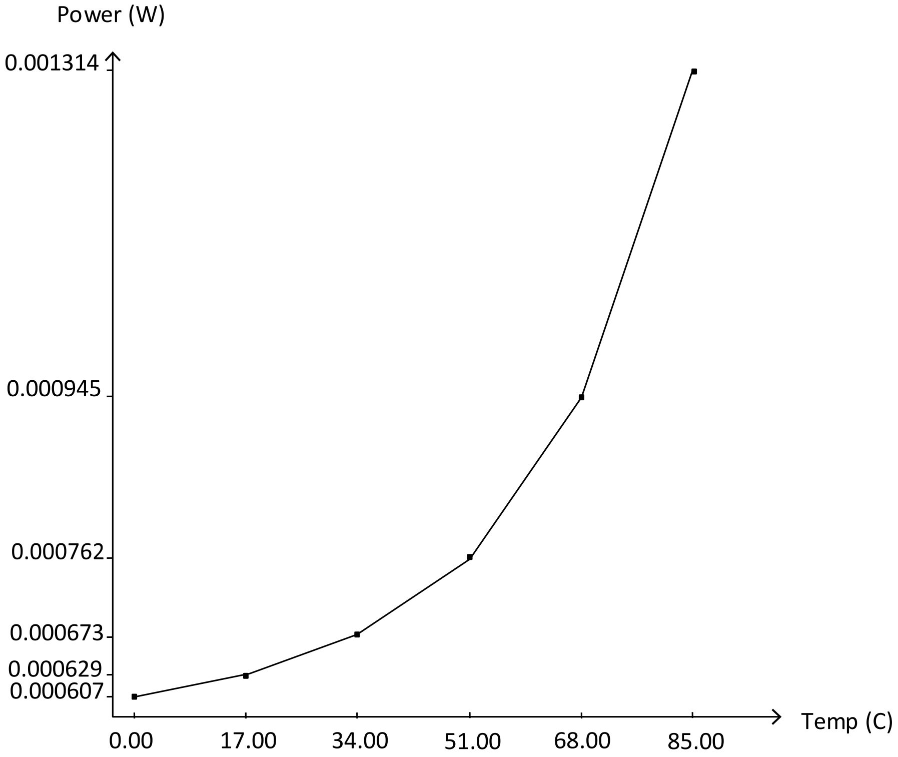


Figure 3.39. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8: Slave Single-Wire Aggregation Device – Total Power versus Ambient Temperature (in °C)

3.5.5. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8 Typical Propagation Delay

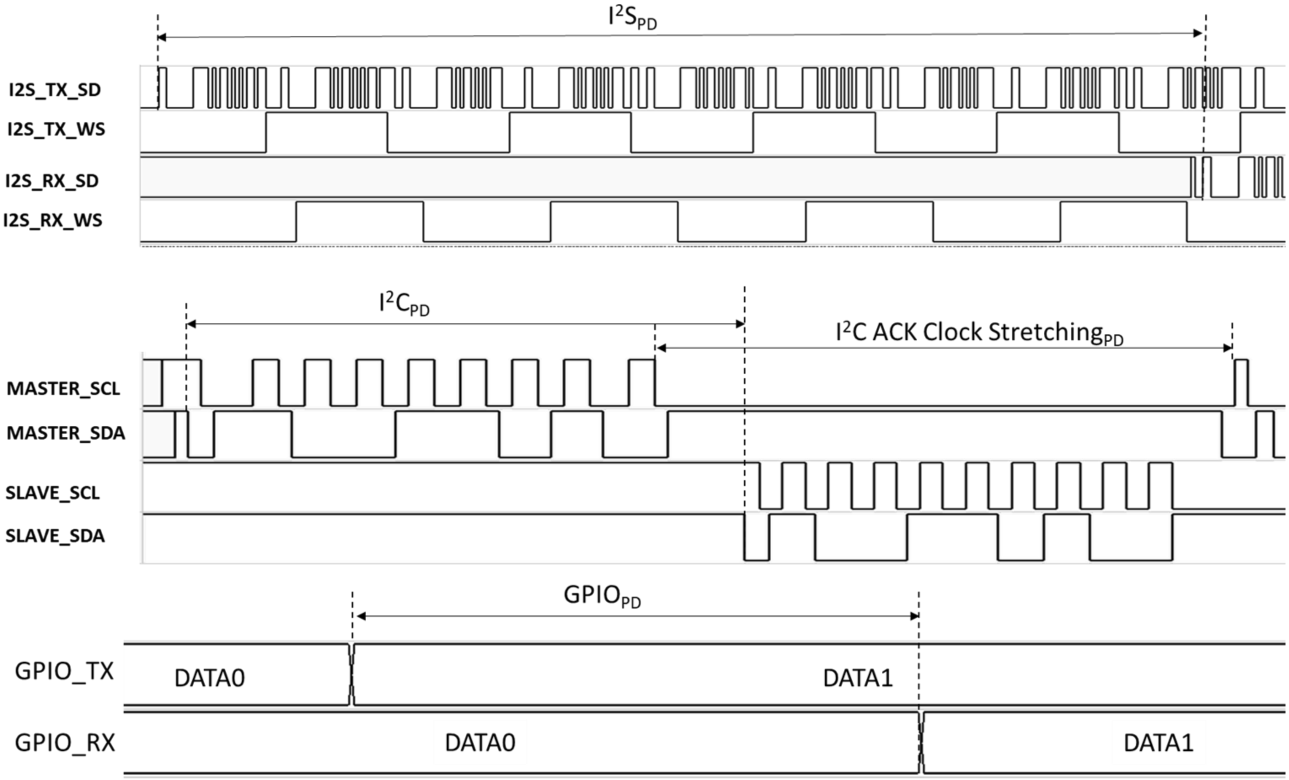


Figure 3.40. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx85 Propagation Delay

Table 3.10. Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx85 Typical Propagation Delay

	Propagation Delay (us)
I^2C_{PD} (SCL Clock at 400 kHz)	26
I^2C ACK Clock Stretching $_{PD}$	24
$GPIO_{PD}$	3.8
$I2S_{PD}$	130

4. Functional Description

This device is initially architecturally based on iCE40 UltraPlus™ FPGA chip, but is designed to be easily portable to other iCE40 and Lattice FPGAs. Its functional description is based on the Single-Wire Signal Aggregation Reference Design (FPGA-RD-02039). This device can take up to seven TX/RX channels to aggregate and communicate over a Single-Wire between two Single-Wire aggregation devices (denoted as Master Single-Wire Aggregation Device and Slave Single-Wire Aggregation Device). The two devices have to be properly configured in terms of number of channels, data content on specified channels, and data width to transmit and retrieve proper information. The single-wire link must be pulled up by strong external resistor, for example, 910 Ω. The 12 MHz clock must be fed to EXT_CLK, which is geared up to ~60 MHz by the on-chip PLL. This ~60 MHz clock is used as a sampling clock on the RX side and ~15 MHz clock is used as TX clock. Two TX clock cycles are required for bi-phase mark coding to transmit one-bit data. Therefore, the transmission data rate is ~7.5 Mbps. The device assumes a ~±20 % clock frequency difference between the externally supplied clock on the master versus slave device.

The Single-Wire Link must have a strong pull-up resistor since it is not driven high for the whole logic *'1'* period. The FPGA drives the link low for the entire *'0'* period, but drives high only for a short time (< 10 ns of beginning of *'1'* period).

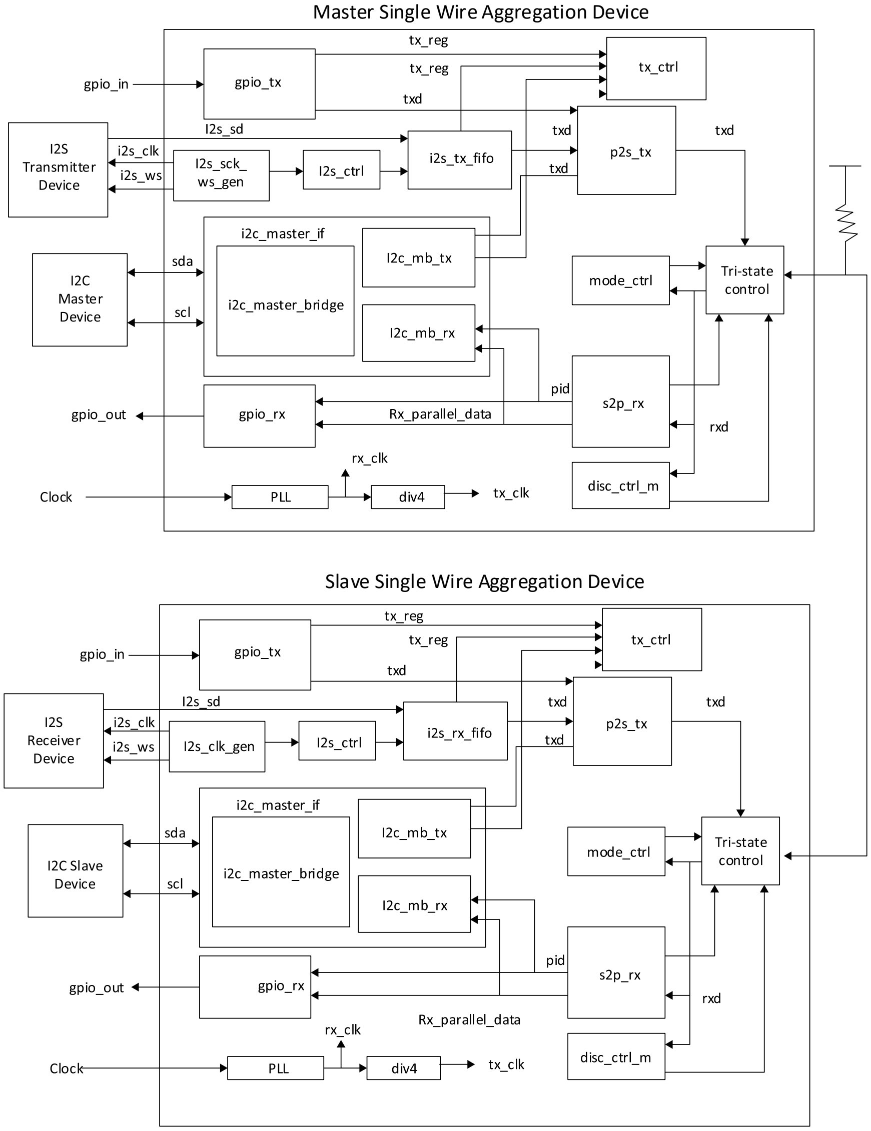


Figure 4.1. Functional Block Diagram

4.1. Link Establishment upon Power and Reset Release

When the devices are powered up and reset is released after configuration, the PLL generates ~60 MHz clock using the 12 MHz input clock at EXT_CLK. This clock is divided by 4 to provide TX clock of ~15 MHz. Figure 4.2 shows the transactions for Link Establishment. After the PLL is locked and a proper TX clock is generated, the Master Single-Wire Aggregation device pulls the link low for 3 TX clock cycles to be discovered by the Slave Single-Wire Aggregation device. It repeats this every 32 clock cycles until it detects 5 cycles or more of link = 0 as a sign of connection acknowledgement. Upon reset release, the Slave device waits for link = 0 for 2 TX clock cycles long, then pulls the link low for 7 TX clock cycles.

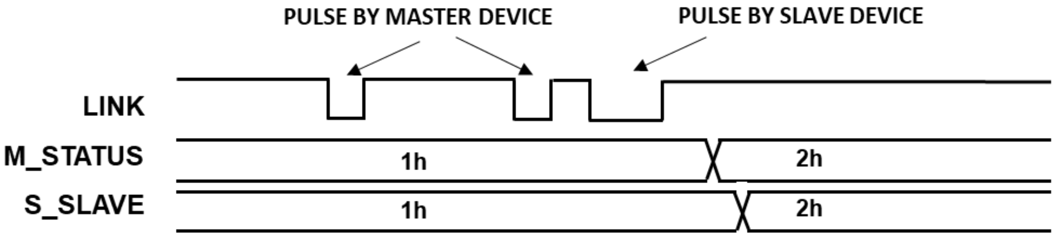


Figure 4.2. Link Establishment

For Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8 and Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8, wherein I2S channels are available, I2S clock learning/training is performed after link acknowledgement. To achieve this, the Master Single-Wire Aggregation Device sends six-eight I2S SCK pulses on the link, which the Slave Single-Wire Aggregation Device receives and processes as shown in Figure 6.3.

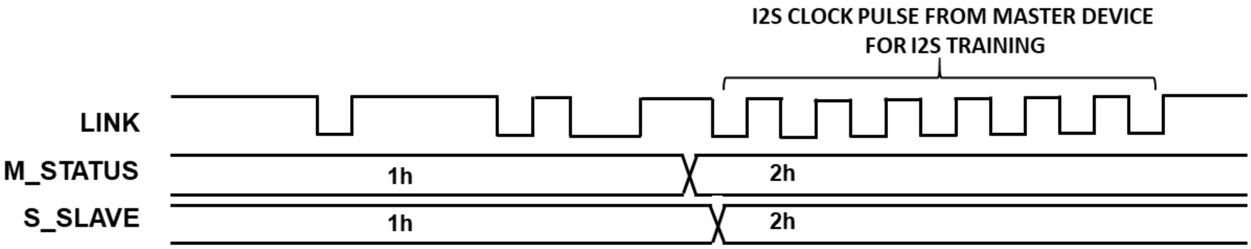


Figure 4.3. Link Establishment

4.2. Link Status

Single-wire Link provides three-bit status outputs to indicate seven conditions. These 3 bits status are connected to drive LED RGB2, RGB1, RGB0 (Status [2:0]):

- 000: Powered up with RESET_N = 0;
- 001: Discovery Stage in which the device is trying to establish a Single-Wire Link connection.
- 010: Connected Stage, link connection established by the pulse exchange shown in the Link Establishment upon Power and Reset Release section.
- 011: Active Stage indicates active payload data transmission on the Single-Wire Link.
- 100: This indicates the device filed to establish TX data on the link.
- 110: This indicates the retransmitting of the payload data on the Link
- 111: This indicates a Parity Error on the payload received.

4.3. TX Rights Negotiation

Figure 4.4 shows an example of TX rights negotiation between two Single-Wire Aggregation Devices. After the link connection is confirmed, both sides can request a TX transaction by pulling the link low for two TX clock cycles. The other side pulls the link low for five TX clock cycles as a grant. If both sides send TX requests at the same time, the long pulse cannot be detected on both sides. In that case, the side previously on RX side gets the TX right and send aTX request pulse again. The other side does not send the TX request pulse again, and waits for a TX request pulse coming from the other side, then sends a grant pulse. If this case happens in the very first transaction after reset release, the Slave Single-Wire Aggregation Device gives up sending a new TX request pulse and the Master Single-Wire Aggregation Device acquires TX rights.

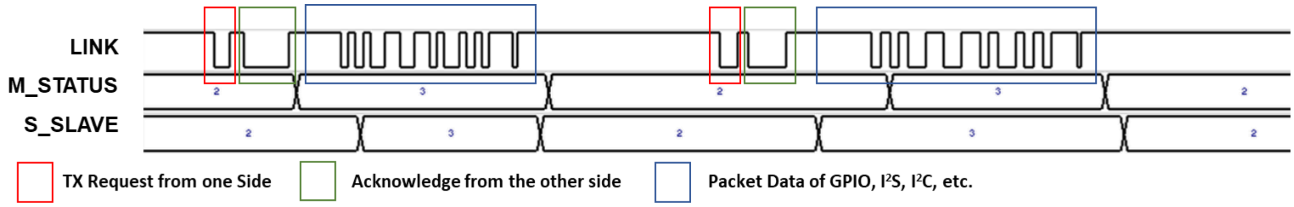
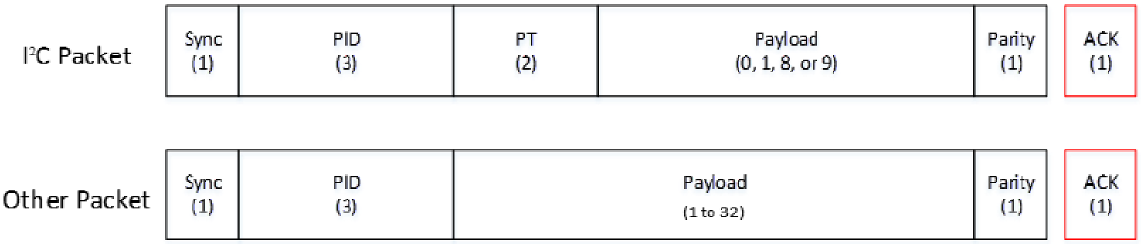

LINK
M_STATUS
S_SLAVE
2
3
2
3
2
2
3
2
3
2
TX Request from one Side
Acknowledge from the other side
Packet Data of GPIO, I²S, I²C, etc.

Figure 4.4. TX Right Negotiation and Packet Transmission

4.4. Packet Transmission

Single-wire employs packet-based TDM data transmission. Figure 4.5 shows a packet structure. Every packet has a start bit, payload ID (PID), and a parity bit. The length of the payload data depends on the PID, Payload Type (in case of I^{2}C), and data width (in case of GPIO). Packet structure is different between I^{2}C, I2S and GPIO. In the case of I^{2}C, 2-bit payload type (PT) indicates the type of payload, since those payloads have different data lengths. PID assignments have to be matched between both Single-Wire Aggregation Devices, otherwise, the RX side cannot retrieve the correct data. These assignments are compile options and cannot be changed dynamically. Parity polarity is determined by the payload length to end the parity bit as high all the time. After the completion of the packet transmission, the RX side returns a short pulse, 4 cycles of Rx clock, as an acknowledge bit (ACK) to notify Parity check is OK. The TX side retransmits the same packet data again if it does not receive an ACK from the RX side. Please note for I2S data, retransmitting the data is not possible.

Bi-phase mark encoding is used to transmit the packet data. Figure 4.6 shows an example of Bi-phase mark encoding and Figure 4.7 shows an example of the bit pattern of an I^{2}C packet. The link status is always high in the idle state. Therefore, the Sync bit, data = 1, is always encoded as 01 followed by PID data. Even parity is used when the number of payload bit is even. Odd parity is used when the number of payload bits is odd. Using this method, the parity bit pattern is either 01 or 11, so the ACK bit is easily recognizable on the TX side. Two TX cycles are assigned to detect the ACK bit on the TX side considering the clock phase difference and frequency tolerance between two Single-Wire aggregation Devices.



Remarks:

- ( ) denotes bit length.
- PID : Payload ID, PID = 7 is reserved and cannot be used for customer purposes.
- PT : Payload Type for I^{2}C
  Mater-to-Slave --- 00 : Start/Repeated start with byte data(8), 01 : write data(8), 10 : ACK/NACK bit(1), 11 : stop (0)
  Slave-to-Master --- 00 : ACK + read data (9), 01 : read data (8), 10 : ACK bit (1), 11 : reserved
- Payload length in non I^{2}C packet is pre-determined by the data width associated with PID.
- Parity : Even Parity is used when payload length is even; Odd Parity is used when payload length is odd
- ACK : ACK bit is returned from RX side to TX side when Parity Check on Rx side is OK. TX side retransmits the same packet if it does not receive ACK bit.

Figure 4.5. TX Right Negotiation and Packet Transmission

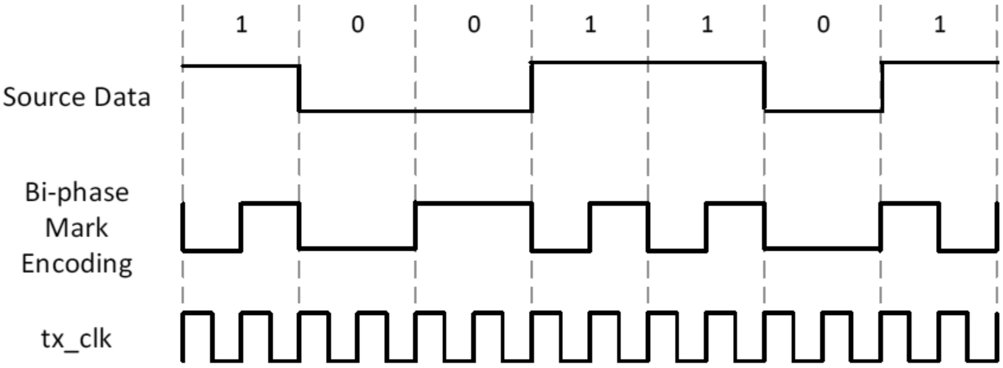


Figure 4.6. TX Right Negotiation and Packet Transmission

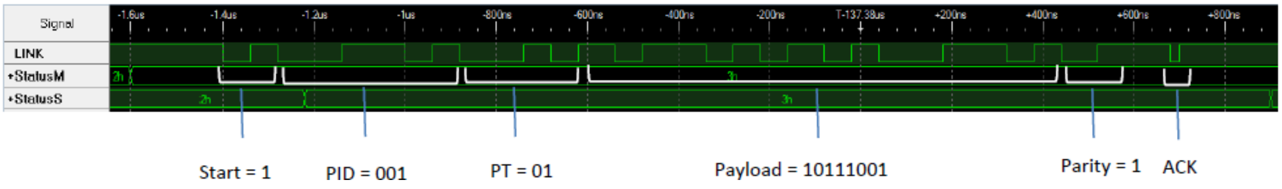


Figure 4.7. Example of I^{2}C Packet

4.5. TX Rights Release

TX side can send the packet of all channels if those are ready to be sent once the device obtains the TX rights. Starting from I2S (Starting from Channel 1 to Channel 2), I^{2}C (Starting Channel 1 to higher channel) and GPIO (Starting Channel 1 to higher channel), it keeps sending packets one after another until all available TX channel data are sent. After that, that device releases TX rights. Therefore, a new negotiation is necessary when it needs to send the next data. RX side sets the waiting period after it returns ACK for the current TX data reception. If it does not receive the start bit within that period and has the internal TX requests, it sends TX request to the other side.

***Note:** Depends on the availability of the said channel in the configuration.

4.6. System Level I^{2}C Transactions

Figure 4.8 and Figure 4.9 show an example of system-level I^{2}C transactions. Two I^{2}C master devices are connected to the Master Single-Wire Aggregation device, such as SCL1M/SDA1M and SCL2M/SDA2M, and two I^{2}C slave devices are connected to the Slave Single-Wire Aggregation device, such as SCL1S/SDA1S and SCL2S/SDA2S. In Figure 4.8, both I^{2}C masters issue Start command followed by I^{2}C address 0x60 and write commands. Then SCL is pulled low by Master Device, while Start Command + I^{2}C address + write command is forwarded to I2C slave device through the link. The Slave Device, which makes I^{2}C ACK from I^{2}C slave device forwarded to theI^{2}C master device through Slave device, link, and Master Single-Wire Aggregation device. In other words, the I^{2}C master device SCL is held low after the I^{2}C master sends a byte of data until the I^{2}C ACK comes back from the other end through the link for write transactions. Figure 4.9 shows Repeated Start command and read transactions. In case of read transaction, master I^{2}C SCL is held low until the Master Single-Wire Aggregation device gets I^{2}C ACK+ read data from the slave side through the link. Since both sides have to replicate the transactions originated from the other sides, I^{2}C transactions take at least twice the time compared to non-aggregated configuration. The actual overhead of Single-wire depends on other link transactions.

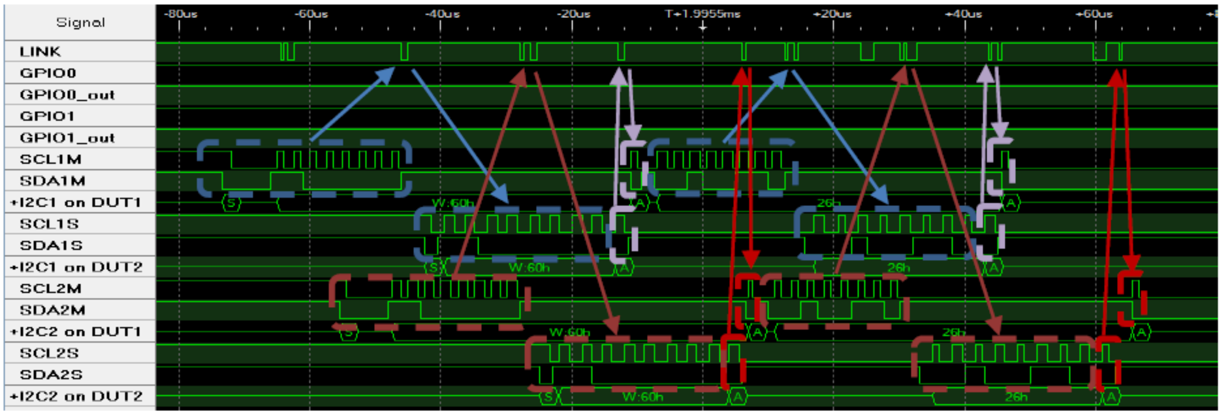


Figure 4.8. I^{2}C Transaction #1 (Sub-address Write for Read Transaction)

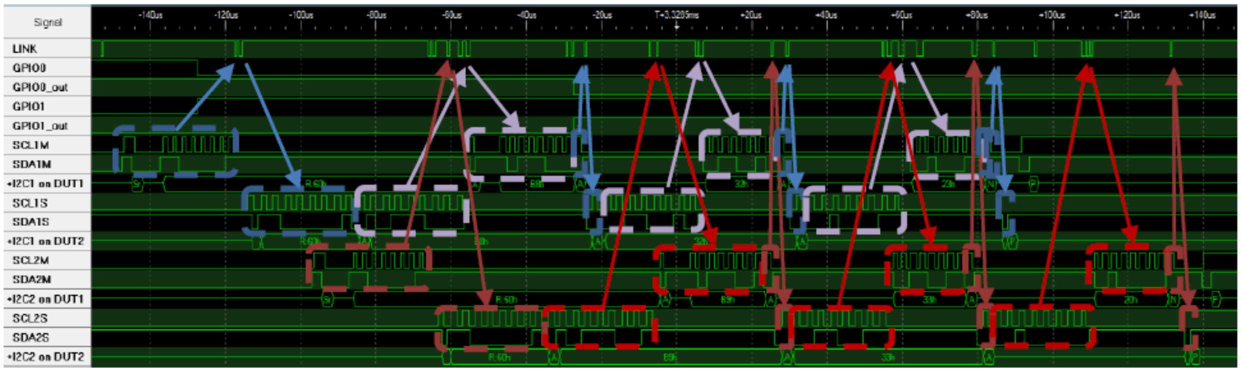


Figure 4.9. I^{2}C Transaction #2 (Repeated Start Followed with Read Transaction)

Figure 4.10 and Figure 4.11 show examples of link delay in case of I^{2}C Start and I^{2}C ACK. Actual delay time depends on several conditions including data sample timing, TX request collision, link occupancy, TX queue, and others.

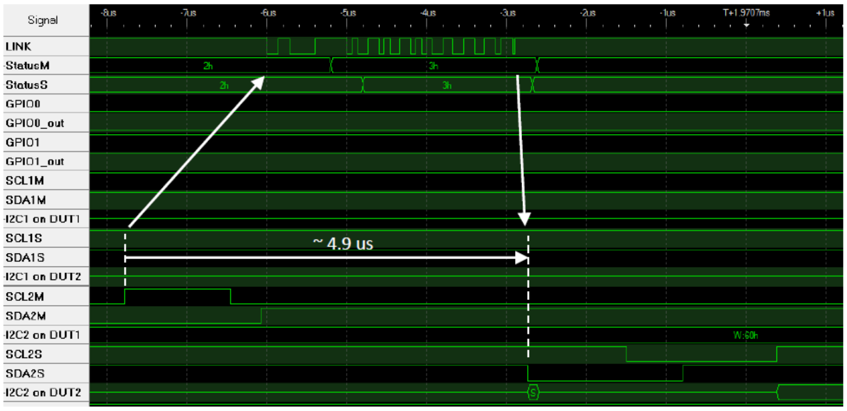


Figure 4.10. Link Delay Example #1 (I^{2}C Start)

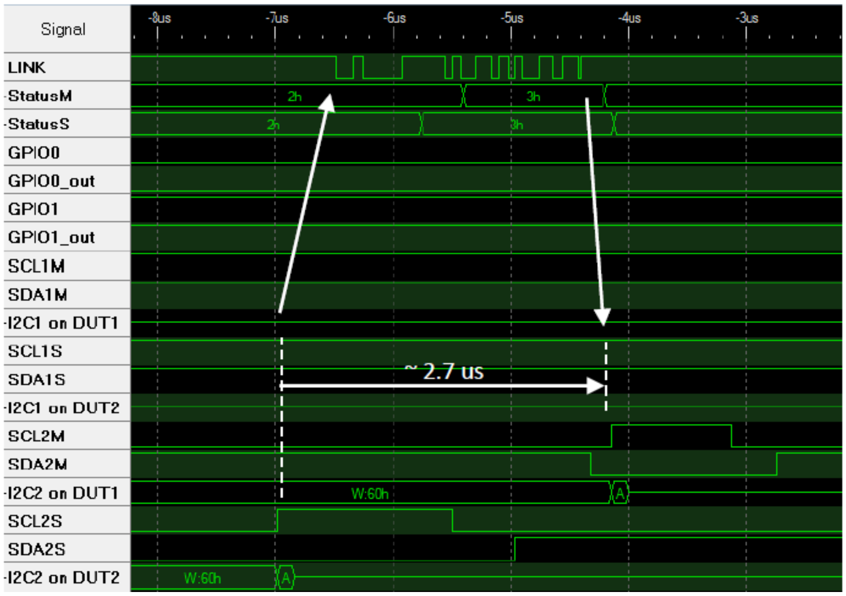


Figure 4.11. Link Delay Example #2 (I^{2}C ACK)

4.7. System Level I2S Transactions

Figure 4.12 shows an example of a system-level I2S transaction. One I2S Transmitter is connected to the Master Single-Wire Aggregation Device while its I2S Receiver is connected to the Slave Single-Wire Aggregation Device. I2S data are sent to the Single-Wire link every Word line, which correspond to the I2S WS. If there are other types of data (I^{2}C for example) connected in the system, data transmission is handled in a round robin manner. For example, in Figure 4.12, the I^{2}C Packet is sent to the Single-Wire after the first packet of I2S data is sent.

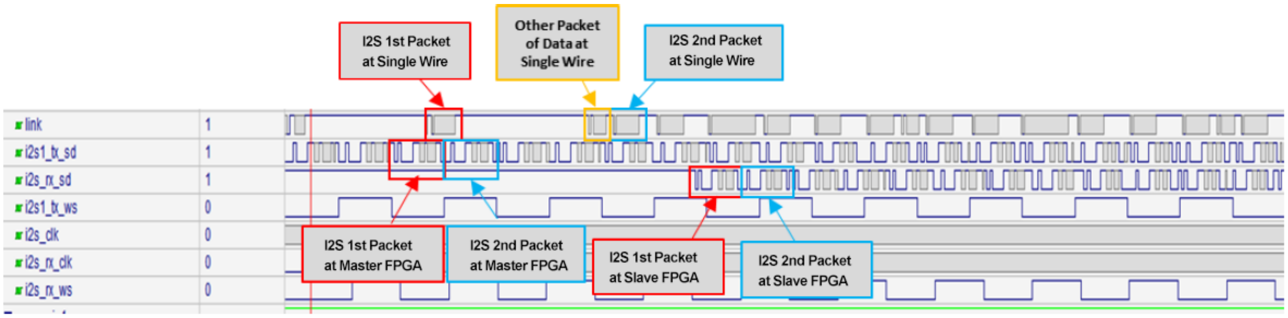


Figure 4.12. System Level I2S Transaction

Figure 4.13 shows an example of I2S delay from Master Single-Wire Aggregation Device to Slave Single-Wire Aggregation Device with a sample rate of 48 kHz, using 32 bit I2S word length. Delay may vary according to the configuration of other data on the link.

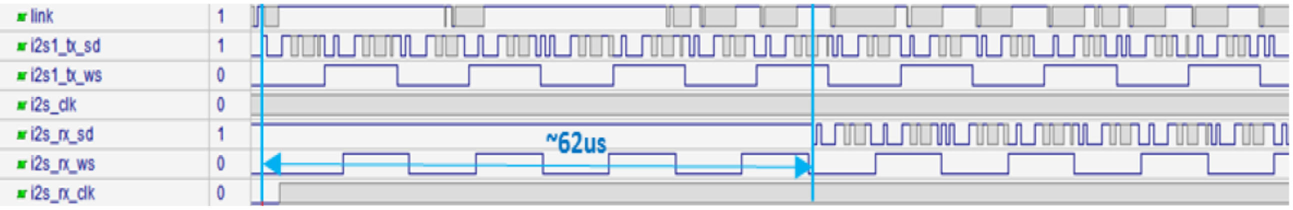


Figure 4.13. I2S Delay from Master to Slave Single-Wire Aggregation Device

For configuration with I2S Channel (Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8 and Configuration I2CMx1_I2CMx1_I2CSx1_GPIOx8), both the Master Single-Wire Aggregation Device and the Slave Single-Wire Aggregation device include I2S FIFO buffer (with FIFO depth at 6). This is to queue valid I2S data on TX side when the Single-Wire Link is busy for I2S payload to be transmitted on the Link and to regenerate the I2S data on the RX side. When I2S FIFO Buffer on the RX side is near empty with a valid I2S data, generated I2S clock on the RX side slows down to offset the delay.

4.8. System Level GPIO Transactions

Figure 4.14 shows an example of a system-level GPIO transaction. 12 bits of GPIO Transmitter are connected on the Master Single-Wire Aggregation Device while 12 bits of GPIO Receiver are connected on the Slave Single-Wire Aggregation Device. GPIO transmission is based on the event on the GPIO_TX. Once event on GPIO_TX is observed, it queues for transmission on the Single-Wire Link. The data is deaggregated on the RX side. In a case where there are valid I2S and I^{2}C data ready to be transmitted on the Link on the TX side, I2S data is sent first on the Link follow by I^{2}C and lastly the GPIO. Figure 4.15 shows an example of GPIO delay with 12 width of GPIO data. Delay may vary according to the other signals attached on the link.

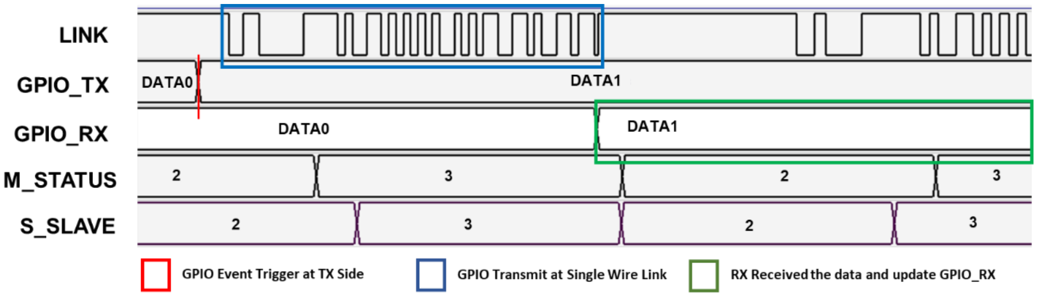

LINK
GPIO_TX
DATA0
DATA1
GPIO_RX
DATA0
DATA1
M_STATUS
2
3
2
3
S_SLAVE
2
3
2
3
GPIO Event Trigger at TX Side
GPIO Transmit at Single Wire Link
RX Received the data and update GPIO_RX

Figure 4.14. GPIO Transaction

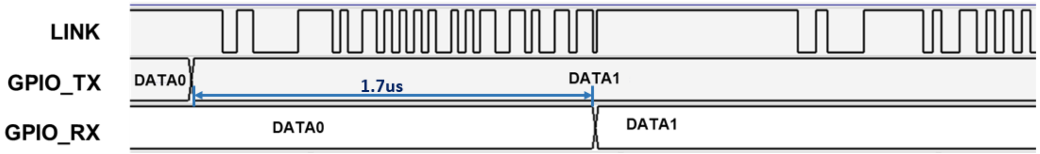


Figure 4.15. 12 Bits Width GPIO Delay from Master to Slave Single-Wire Device

This solution uses the iCE40 UltraPlus device. To know the DC and Switching Characteristic, refer to the iCE40 UltraPlus Family Data Sheet (FPGA-DS-02008).

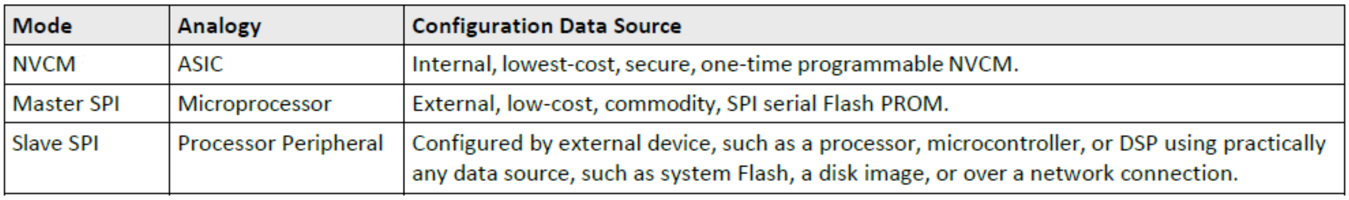
5. Programming and Configuration

The Single-Wire Aggregation devices are iCE40 UltraPlus SRAM-based FPGA. This device has an on-chip, one-time programmable NVCM (Non-Volatile Configuration Memory) to store configuration data. The SRAM memory cells are volatile, meaning that once power is removed from the device, its configuration is lost, and must be reloaded on the next power-up. This behavior has the advantage of being re-programmable in the field, which provides flexibility for products already deployed to the field, but it also requires that the configuration information be stored in a non-volatile device and loaded each time power is applied to the device. The on-chip NVCM allows the device to configure instantly and enhances the design security by eliminating the need to use an external memory device. The configuration data can also be stored in an external SPI Flash from which the FPGA can configure itself upon power-up. This is useful for prototyping the FPGA or in situations where re-configurability is required. Additionally, the device can be configured by a processor in an embedded environment.

As described in Table 5.1, the Single-Wire Aggregation Device components are configured for a specific application by loading a binary configuration bitstream image generated by the Lattice Radiant Software design tool. For high-volume applications, the bitstream image is usually permanently programmed in the on-chip Non-volatile Configuration Memory. However, the bitstream image can also be stored externally in a standard, low-cost commodity SPI serial Flash PROM. The device can automatically load the image using the SPI Master Configuration Interface. Similarly, the device's configuration data can be downloaded from an external processor, microcontroller, or DSP using SPI serial interface.

For more details on configuring the iCE40 UltraPlus, refer to iCE40 Programming and Configuration (TN1248).

Table 5.1. Configuration Options.



Mode	Analogy	Configuration Data Source
NVCM	ASIC	Internal, lowest-cost, secure, one-time programmable NVCM.
Master SPI	Microprocessor	External, low-cost, commodity, SPI serial Flash PROM.
Slave SPI	Processor Peripheral	Configured by external device, such as a processor, microcontroller, or DSP using practically any data source, such as system Flash, a disk image, or over a network connection.

5.1. Bitstream for Ready to Use Device Configuration

Ready to Use Device Configuration for Single-Wire Aggregation Device is available on www.latticesemi.com. Figure 5.1 shows the directory structure and where the Ready to use available bitstreams can be located. Table 5.2 shows the valid bitstream combination for ready to use configurations. Aside for the ready to use configurations, SWA demo bitstreams are also included on the folder.

Table 5.2. Valid Bitstream Combination for Ready to Use Configuration

Configuration	Bitstream File for Single-Wire Aggregation Master Device	Bitstream File for Single-Wire Aggregation Slave Device
I2Sx2_I2CSx1_I2CMx1_GPIOx8	I2Sx2_I2CSx1_I2CMx1_GPIOx8 _Master.bin	I2Sx2_I2CSx1_I2CMx1_GPIOx8 _Slave.bin
I2CMx6_GPIOx2	I2CMx6_GPIOx2 _Master.bin	I2CMx6_GPIOx2 _Slave.bin
I2CMx1_GPIOx12	I2CMx1_GPIOx12 _Master.bin	I2CMx1_GPIOx12 _Slave.bin
I2CMx3_I2CSx2_GPIOx15	I2CMx3_I2CSx2_GPIOx15 _Master.bin	I2CMx3_I2CSx2_GPIOx15 _Slave.bin
I2Sx1_I2CMx1_I2CSx1_GPIOx8	I2Sx1_I2CMx1_I2CSx1_GPIOx8_Master.bin	I2Sx1_I2CMx1_I2CSx1_GPIOx8 _Slave.bin
Configuration Demo	ConfigurationDemo_Master.bin	ConfigurationDemo_Slave.bin

5.2. Non-Volatile Configuration Memory

The NVCM is a One Time Programmable (OTP) memory and is large enough to program the Single-Wire Aggregation device. It provides the capability for iCE devices to perform in a stand-alone mode, essentially behaving like an ASIC. The NVCM memory also has very high programming yield due to extensive error checking and correction (ECC) circuitry. The NVCM is ideal for cost-sensitive, high-volume production applications, saving the cost and board space associated with an external configuration PROM. Furthermore, the NVCM provides design security, protecting critical intellectual property (IP). The NVCM contents are entirely contained within the device and are not readable once protected by the one-time programmable Security bits. Furthermore, there is no observable difference between a programmed and unprogrammed memory cell using optical or electron microscopy.

The NVCM memory has a programming interface similar to a 25-series SPI serial Flash PROM. Consequently, it can be programmed using Lattice Radiant Programmer (version 1.1 or later) before or after circuit board assembly or programmed in-system from a microprocessor or other intelligent controller. The NVCM can also be pre-programmed at the factory. Contact Lattice Technical Support or your local Lattice sales office for assistance.

5.2.1. NVCM Programming

The NVCM can be programmed in the following ways:

- Diamond Programmer
  - Programming using the Lattice Radiant Programmer (Lattice Radiant 1.1 or later) is recommended for prototyping.
  - Programming is supported using the Lattice programming cable. For more information refer to the Diamond Programmer Online Help and Programming Cables User Guide (FPGA-UG-02042).
- Factory Programming
  - The Lattice factory offers NVCM programming. For more information contact your local Lattice sales office.
- Embedded Programming. The NVCM can be programmed using a processor. For more information, contact your local Lattice sales office.

5.2.2. SPI Master Configuration Interface

The device can be configured from an external, commodity SPI serial Flash PROM, as shown in Figure 5.1. The SPI configuration interface is essentially its own independent I/O bank, powered by the VCC_SPI supply input. Presently, most commercially available SPI serial Flash PROMs require a 3.3 V supply.

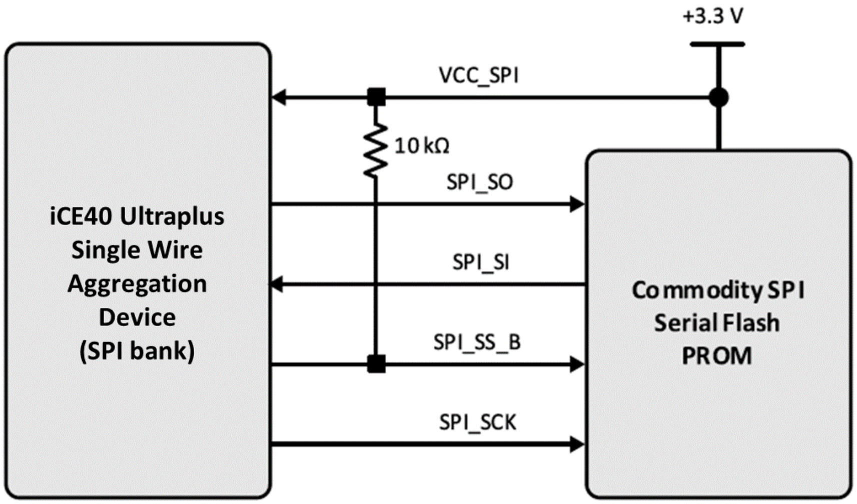
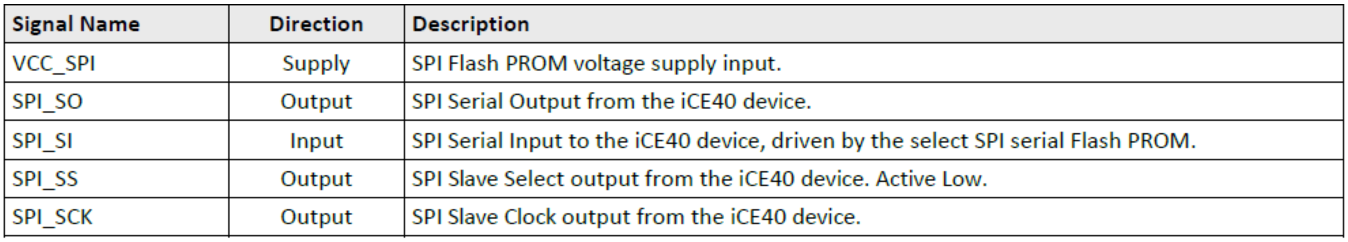

+3.3 V
VCC_SPI
10 kΩ
SPI_SO
SPI_SI
SPI_SS_B
SPI_SCK
iCE40 Ultraplus Single Wire Aggregation Device (SPI bank)
Commodity SPI Serial Flash PROM

Figure 5.1. SPI Master Configuration Interface

The SPI configuration interface is used primarily during development before mass production, where the configuration is then permanently programmed in the NVCM configuration memory (only available in iCE40 LP, iCE40 HX, iCE40 Ultra, iCE40 UltraLite and iCE40 UltraPlus devices). However, the SPI interface can also be the primary configuration interface allowing easy in-system upgrades and support for multiple configuration images.

The SPI control signals are defined in Table 5.3.

Table 5.3. Configuration Options



Signal Name	Direction	Description
VCC_SPI	Supply	SPI Flash PROM voltage supply input.
SPI_SO	Output	SPI Serial Output from the iCE40 device.
SPI_SI	Input	SPI Serial Input to the iCE40 device, driven by the select SPI serial Flash PROM.
SPI_SS	Output	SPI Slave Select output from the iCE40 device. Active Low.
SPI_SCK	Output	SPI Slave Clock output from the iCE40 device.

For more details on configuring the SPI Master Configuration Mode, refer to iCE40 Programming and Configuration (TN1248). The NVCM can be programmed using a processor. For more information, contact your local Lattice sales office.

5.2.3. Device Configuration

There are various ways to configure the Configuration RAM (CRAM) using the SPI port, including:

- From a SPI Flash (Master SPI mode)
- System microprocessor to drive a Serial Slave SPI port (SSPI mode)

For more details on configuring the iCE40 UltraPlus, refer to iCE40 Programming and Configuration (TN1248).

6. Advance Reconfiguration Options

Single-Wire Aggregation device is based in iCE UltraPlus FPGA device. This allows you to modify the ready-to-use configuration according to their system level requirement. These changes include:

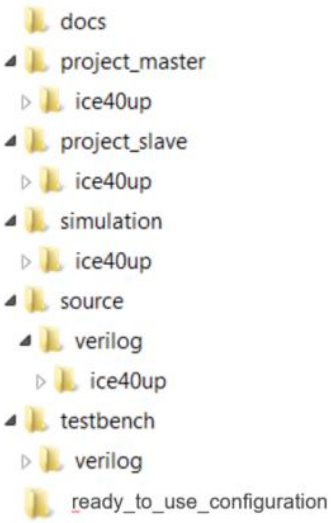
- Change of Pin assignments.
- Change of I^{2}C clock rate
- Change of I2S sampling rate and sampling width
- Change of GPIO data width.
- Change number of I2S, I^{2}C and GPIO channels.

Advance reconfiguration is applicable for users with experience in RTL design using Lattice Radiant Software. For new configuration requests, contact Lattice Technical Support or your local Lattice sales office for assistance.

6.1. Packaged Design

The Lattice Single-Wire Aggregation device is based on Lattice Single-wire Aggregation Reference Design (FPGA-RD-02039). Figure 9.1 shows the directory structure.

There are two projects for Single-Wire Aggregation Master and Slave Device. These projects can be opened using Lattice Radiant Software (version 1.1 or later). To open the Single-Wire Aggregation Master projects, launch Lattice Radiant Software. Then, open the project under project_master/ice40up/project_master.rdf. Single-Wire Aggregation Slave Project can be found in project_slave/ice40up/project_slave.rdf.



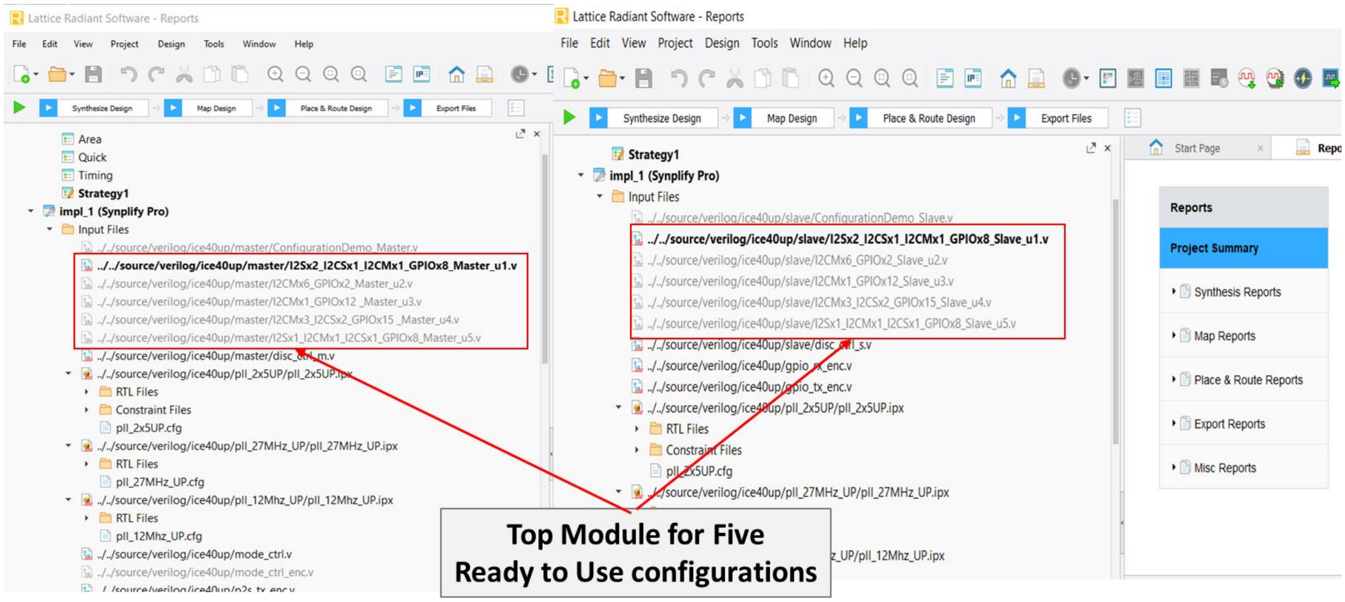
```
  docs
◢ project_master
  ▷ ice40up
◢ project_slave
  ▷ ice40up
◢ simulation
  ▷ ice40up
◢ source
  ◢ verilog
    ▷ ice40up
◢ testbench
  ▷ verilog
  ready_to_use_configuration
```

Figure 6.1. Packaged Design Directory Structure

Figure 6.2 shows the opened project for master and slave. The project has five top modules named as singlewire_master/slave_u[1, 2, 3, 4, 5].v. These top modules correspond to the five top modules used by the Ready-to-Use Configurations. Similarly, Figure 6.3 shows the five Pins Constraints Files, which also correspond to the Pins Constraints Files used by the Ready-to-Use Configurations. This is the same for both master and slave projects.

To modify a particular configuration, only one top module and pins constraints file should be active. Figure 6.4 and Figure 6.5 shows how to include and exclude Top module on the implementation. In addition, Figure 6.6 shows how to activate the targeted Pins Constraints File. Ensure that other top modules and Pins constraints are excluded on the project.

Modifying the Top module parameters and port list are similar to Lattice Single-wire Aggregation Design (FPGA-RD-02039).

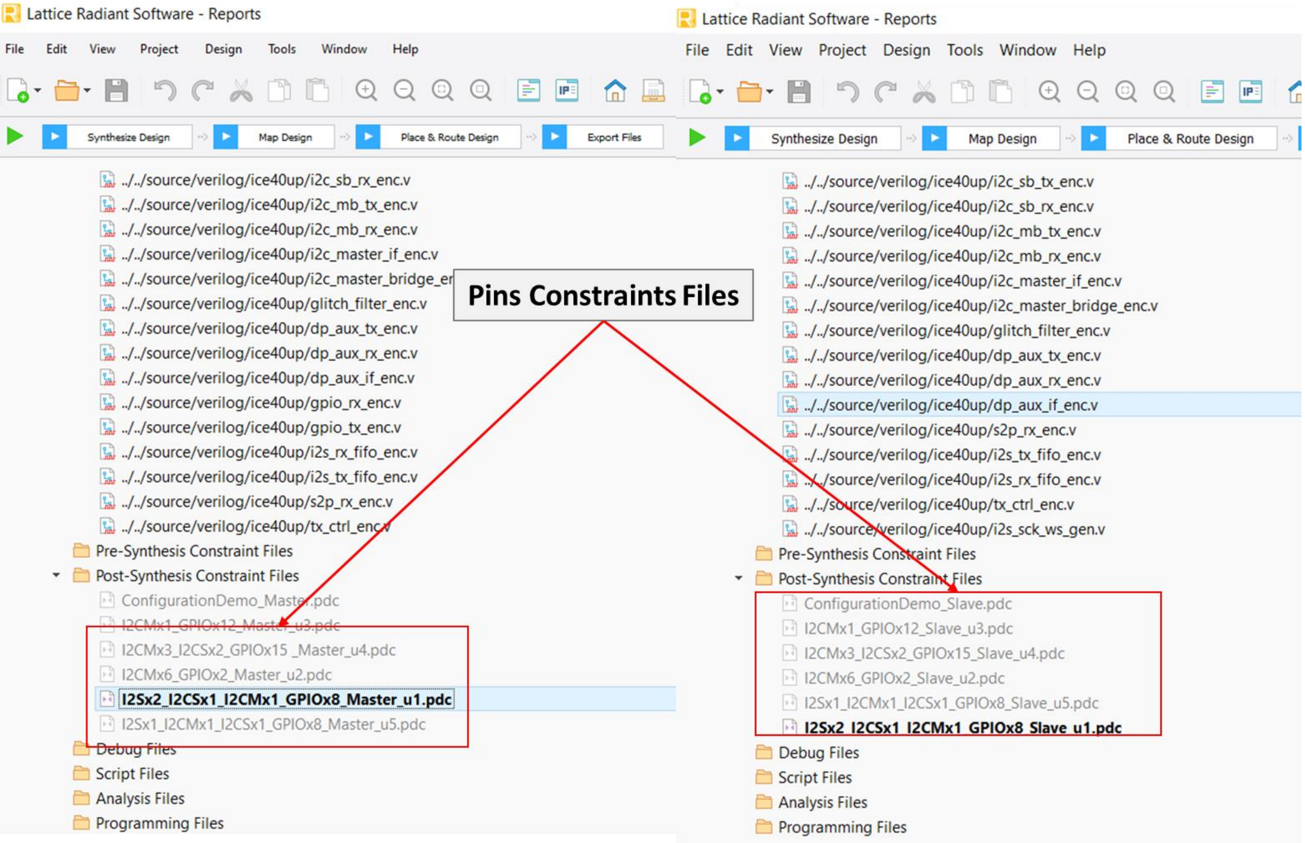


```
Area
Quick
Timing
Strategy1
impl_1 (Synplify Pro)
  Input Files
    ../../source/verilog/ice40up/master/ConfigurationDemo_Master.v
    ../../source/verilog/ice40up/master/I2Sx2_I2CSx1_I2CMx1_GPIOx8_Master_u1.v
    ../../source/verilog/ice40up/master/I2CMx6_GPIOx2_Master_u2.v
    ../../source/verilog/ice40up/master/I2CMx1_GPIOx12 _Master_u3.v
    ../../source/verilog/ice40up/master/I2CMx3_I2CSx2_GPIOx15 _Master_u4.v
    ../../source/verilog/ice40up/master/I2Sx1_I2CMx1_I2CSx1_GPIOx8_Master_u5.v
    ../../source/verilog/ice40up/master/disc_ctrl_m.v
    ../../source/verilog/ice40up/pll_2x5UP/pll_2x5UP.ipx
      RTL Files
      Constraint Files
      pll_2x5UP.cfg
    ../../source/verilog/ice40up/pll_27MHz_UP/pll_27MHz_UP.ipx
      RTL Files
      pll_27MHz_UP.cfg
    ../../source/verilog/ice40up/pll_12Mhz_UP/pll_12Mhz_UP.ipx
      RTL Files
      pll_12Mhz_UP.cfg
    ../../source/verilog/ice40up/mode_ctrl.v
    ../../source/verilog/ice40up/mode_ctrl_enc.v
    ../../source/verilog/ice40up/p2s_tx_enc.v
```

```
Strategy1
impl_1 (Synplify Pro)
  Input Files
    ../../source/verilog/ice40up/slave/ConfigurationDemo_Slave.v
    ../../source/verilog/ice40up/slave/I2Sx2_I2CSx1_I2CMx1_GPIOx8_Slave_u1.v
    ../../source/verilog/ice40up/slave/I2CMx6_GPIOx2_Slave_u2.v
    ../../source/verilog/ice40up/slave/I2CMx1_GPIOx12_Slave_u3.v
    ../../source/verilog/ice40up/slave/I2CMx3_I2CSx2_GPIOx15_Slave_u4.v
    ../../source/verilog/ice40up/slave/I2Sx1_I2CMx1_I2CSx1_GPIOx8_Slave_u5.v
    ../../source/verilog/ice40up/slave/disc_ctrl_s.v
    ../../source/verilog/ice40up/gpio_rx_enc.v
    ../../source/verilog/ice40up/gpio_tx_enc.v
    ../../source/verilog/ice40up/pll_2x5UP/pll_2x5UP.ipx
      RTL Files
      Constraint Files
      pll_2x5UP.cfg
    ../../source/verilog/ice40up/pll_27MHz_UP/pll_27MHz_UP.ipx
    z_UP/pll_12Mhz_UP.ipx
```

Top Module for Five Ready to Use configurations

Figure 6.2. Single-Wire Aggregation Master and Slave Project

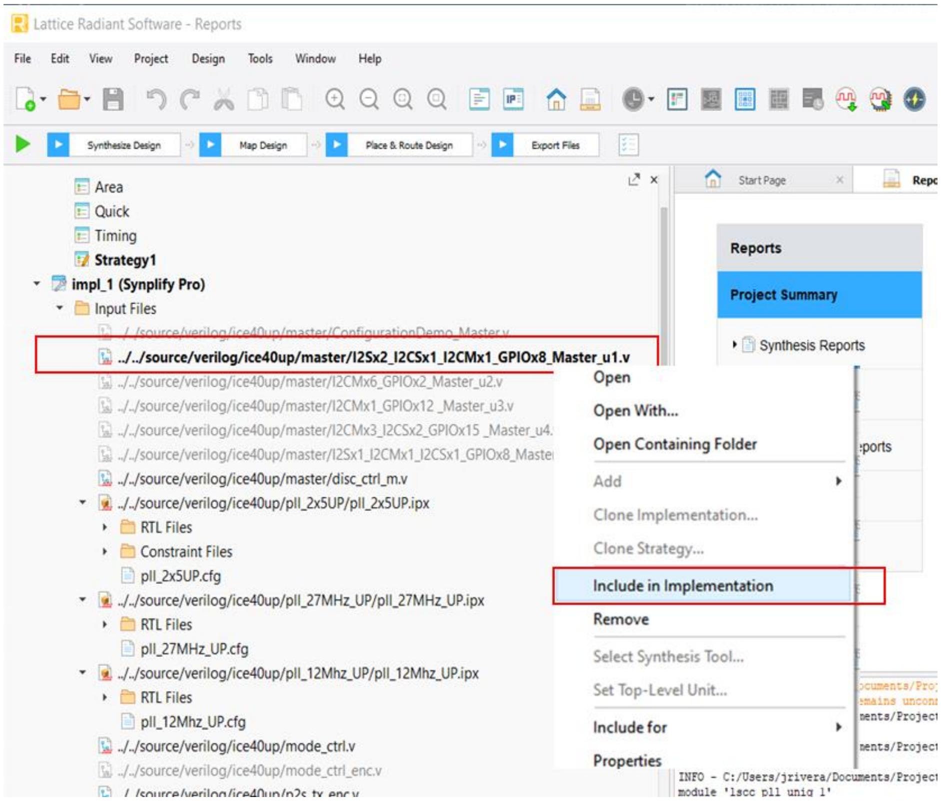

Lattice Radiant Software - Reports
File Edit View Project Design Tools Window Help
Synthesize Design
Map Design
Place & Route Design
Export Files

```
../../source/verilog/ice40up/i2c_sb_rx_enc.v
../../source/verilog/ice40up/i2c_mb_tx_enc.v
../../source/verilog/ice40up/i2c_mb_rx_enc.v
../../source/verilog/ice40up/i2c_master_if_enc.v
../../source/verilog/ice40up/i2c_master_bridge_er
../../source/verilog/ice40up/glitch_filter_enc.v
../../source/verilog/ice40up/dp_aux_tx_enc.v
../../source/verilog/ice40up/dp_aux_rx_enc.v
../../source/verilog/ice40up/dp_aux_if_enc.v
../../source/verilog/ice40up/gpio_rx_enc.v
../../source/verilog/ice40up/gpio_tx_enc.v
../../source/verilog/ice40up/i2s_rx_fifo_enc.v
../../source/verilog/ice40up/i2s_tx_fifo_enc.v
../../source/verilog/ice40up/s2p_rx_enc.v
../../source/verilog/ice40up/tx_ctrl_enc.v
Pre-Synthesis Constraint Files
Post-Synthesis Constraint Files
  ConfigurationDemo_Master.pdc
  I2CMx1_GPIOx12_Master_u3.pdc
  I2CMx3_I2CSx2_GPIOx15 _Master_u4.pdc
  I2CMx6_GPIOx2_Master_u2.pdc
  I2Sx2_I2CSx1_I2CMx1_GPIOx8_Master_u1.pdc
  I2Sx1_I2CMx1_I2CSx1_GPIOx8_Master_u5.pdc
Debug Files
Script Files
Analysis Files
Programming Files
```

```
../../source/verilog/ice40up/i2c_sb_tx_enc.v
../../source/verilog/ice40up/i2c_sb_rx_enc.v
../../source/verilog/ice40up/i2c_mb_tx_enc.v
../../source/verilog/ice40up/i2c_mb_rx_enc.v
../../source/verilog/ice40up/i2c_master_if_enc.v
../../source/verilog/ice40up/i2c_master_bridge_enc.v
../../source/verilog/ice40up/glitch_filter_enc.v
../../source/verilog/ice40up/dp_aux_tx_enc.v
../../source/verilog/ice40up/dp_aux_rx_enc.v
../../source/verilog/ice40up/dp_aux_if_enc.v
../../source/verilog/ice40up/s2p_rx_enc.v
../../source/verilog/ice40up/i2s_tx_fifo_enc.v
../../source/verilog/ice40up/i2s_rx_fifo_enc.v
../../source/verilog/ice40up/tx_ctrl_enc.v
../../source/verilog/ice40up/i2s_sck_ws_gen.v
Pre-Synthesis Constraint Files
Post-Synthesis Constraint Files
  ConfigurationDemo_Slave.pdc
  I2CMx1_GPIOx12_Slave_u3.pdc
  I2CMx3_I2CSx2_GPIOx15_Slave_u4.pdc
  I2CMx6_GPIOx2_Slave_u2.pdc
  I2Sx1_I2CMx1_I2CSx1_GPIOx8_Slave_u5.pdc
  I2Sx2_I2CSx1_I2CMx1_GPIOx8_Slave_u1.pdc
Debug Files
Script Files
Analysis Files
Programming Files
```

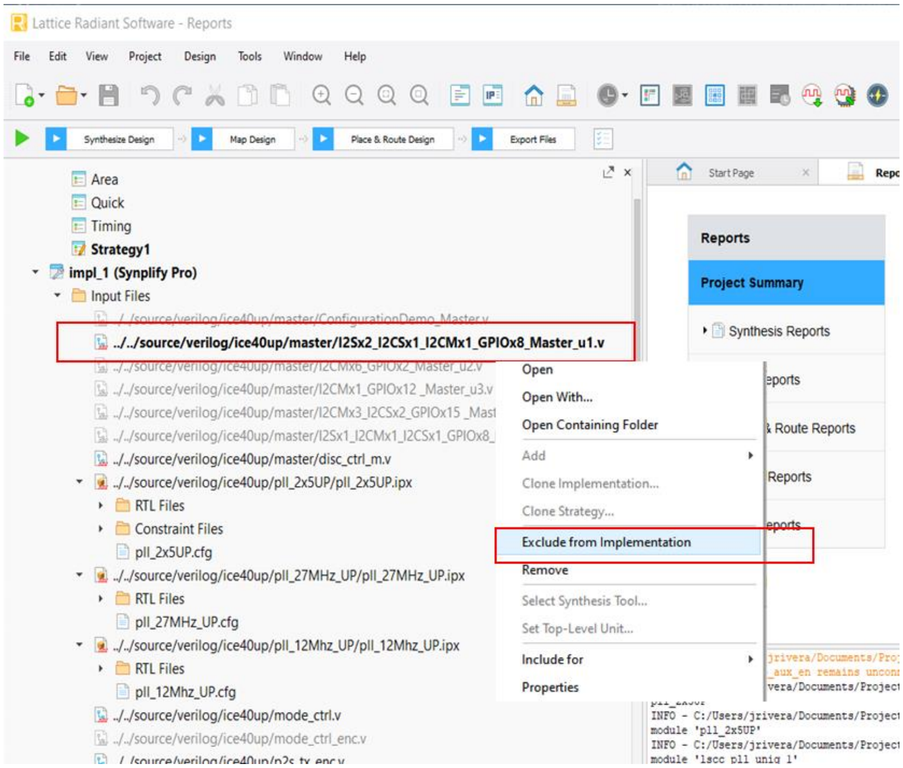
Pins Constraints Files

Figure 6.3. Single-Wire Aggregation Master and Slave Pins Constraints Files


Lattice Radiant Software - Reports
File Edit View Project Design Tools Window Help
Synthesize Design
Map Design
Place & Route Design
Export Files
Area
Quick
Timing
Strategy1
impl_1 (Synplify Pro)
Input Files
../../source/verilog/ice40up/master/ConfigurationDemo_Master.v
../../source/verilog/ice40up/master/I2Sx2_I2CSx1_I2CMx1_GPIOx8_Master_u1.v
../../source/verilog/ice40up/master/I2CMx6_GPIOx2_Master_u2.v
../../source/verilog/ice40up/master/I2CMx1_GPIOx12 _Master_u3.v
../../source/verilog/ice40up/master/I2CMx3_I2CSx2_GPIOx15 _Master_u4
../../source/verilog/ice40up/master/I2Sx1_I2CMx1_I2CSx1_GPIOx8_Master
../../source/verilog/ice40up/master/disc_ctrl_m.v
../../source/verilog/ice40up/pll_2x5UP/pll_2x5UP.ipx
RTL Files
Constraint Files
pll_2x5UP.cfg
../../source/verilog/ice40up/pll_27MHz_UP/pll_27MHz_UP.ipx
RTL Files
pll_27MHz_UP.cfg
../../source/verilog/ice40up/pll_12Mhz_UP/pll_12Mhz_UP.ipx
RTL Files
pll_12Mhz_UP.cfg
../../source/verilog/ice40up/mode_ctrl.v
../../source/verilog/ice40up/mode_ctrl_enc.v
../../source/verilog/ice40up/p2s_tx_enc.v
Open
Open With...
Open Containing Folder
Add
Clone Implementation...
Clone Strategy...
Include in Implementation
Remove
Select Synthesis Tool...
Set Top-Level Unit...
Include for
Properties
Start Page
Repo
Reports
Project Summary
Synthesis Reports
ports

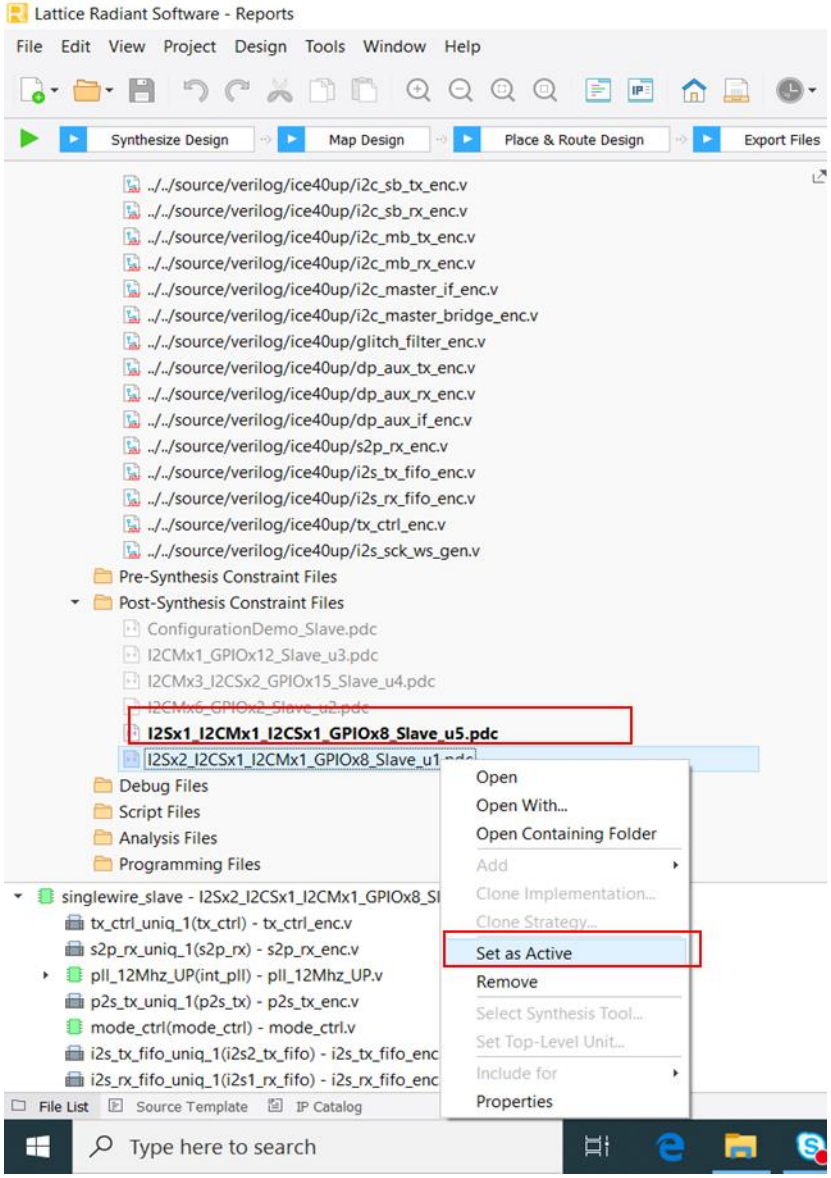
```
INFO - C:/Users/jrivera/Documents/Project
module 'lscc_pll_uniq_1'
```

Figure 6.4. Implementing Top Module – Only One Top Module Implemented


Lattice Radiant Software - Reports
File Edit View Project Design Tools Window Help
Synthesize Design
Map Design
Place & Route Design
Export Files
Area
Quick
Timing
Strategy1
impl_1 (Synplify Pro)
Input Files
../../source/verilog/ice40up/master/ConfigurationDemo_Master.v
../../source/verilog/ice40up/master/I2Sx2_I2CSx1_I2CMx1_GPIOx8_Master_u1.v
../../source/verilog/ice40up/master/I2CMx6_GPIOx2_Master_u2.v
../../source/verilog/ice40up/master/I2CMx1_GPIOx12 _Master_u3.v
../../source/verilog/ice40up/master/I2CMx3_I2CSx2_GPIOx15 _Mast
../../source/verilog/ice40up/master/I2Sx1_I2CMx1_I2CSx1_GPIOx8_
../../source/verilog/ice40up/master/disc_ctrl_m.v
../../source/verilog/ice40up/pll_2x5UP/pll_2x5UP.ipx
RTL Files
Constraint Files
pll_2x5UP.cfg
../../source/verilog/ice40up/pll_27MHz_UP/pll_27MHz_UP.ipx
RTL Files
pll_27MHz_UP.cfg
../../source/verilog/ice40up/pll_12Mhz_UP/pll_12Mhz_UP.ipx
RTL Files
pll_12Mhz_UP.cfg
../../source/verilog/ice40up/mode_ctrl.v
../../source/verilog/ice40up/mode_ctrl_enc.v
../../source/verilog/ice40up/p2s_tx_enc.v
Open
Open With...
Open Containing Folder
Add
Clone Implementation...
Clone Strategy...
Exclude from Implementation
Remove
Select Synthesis Tool...
Set Top-Level Unit...
Include for
Properties
Start Page
Repo
Reports
Project Summary
Synthesis Reports
eports
& Route Reports
Reports
eports

```
INFO - C:/Users/jrivera/Documents/Project
module 'pll_2x5UP'
INFO - C:/Users/jrivera/Documents/Project
module 'lscc_pll_uniq_1'
```

Figure 6.5. Excluding other Top Modules for Implementation – Only One Top Module Implemented



Lattice Radiant Software - Reports

../../source/verilog/ice40up/i2c_sb_tx_enc.v
../../source/verilog/ice40up/i2c_sb_rx_enc.v
../../source/verilog/ice40up/i2c_mb_tx_enc.v
../../source/verilog/ice40up/i2c_mb_rx_enc.v
../../source/verilog/ice40up/i2c_master_if_enc.v
../../source/verilog/ice40up/i2c_master_bridge_enc.v
../../source/verilog/ice40up/glitch_filter_enc.v
../../source/verilog/ice40up/dp_aux_tx_enc.v
../../source/verilog/ice40up/dp_aux_rx_enc.v
../../source/verilog/ice40up/dp_aux_if_enc.v
../../source/verilog/ice40up/s2p_rx_enc.v
../../source/verilog/ice40up/i2s_tx_fifo_enc.v
../../source/verilog/ice40up/i2s_rx_fifo_enc.v
../../source/verilog/ice40up/tx_ctrl_enc.v
../../source/verilog/ice40up/i2s_sck_ws_gen.v
Pre-Synthesis Constraint Files
Post-Synthesis Constraint Files
ConfigurationDemo_Slave.pdc
I2CMx1_GPIOx12_Slave_u3.pdc
I2CMx3_I2CSx2_GPIOx15_Slave_u4.pdc
I2CMx6_GPIOx2_Slave_u2.pdc
I2Sx1_I2CMx1_I2CSx1_GPIOx8_Slave_u5.pdc
I2Sx2_I2CSx1_I2CMx1_GPIOx8_Slave_u1.pdc
Debug Files
Script Files
Analysis Files
Programming Files

singlewire_slave - I2Sx2_I2CSx1_I2CMx1_GPIOx8_Sl
tx_ctrl_uniq_1(tx_ctrl) - tx_ctrl_enc.v
s2p_rx_uniq_1(s2p_rx) - s2p_rx_enc.v
pll_12Mhz_UP(int_pll) - pll_12Mhz_UP.v
p2s_tx_uniq_1(p2s_tx) - p2s_tx_enc.v
mode_ctrl(mode_ctrl) - mode_ctrl.v
i2s_tx_fifo_uniq_1(i2s2_tx_fifo) - i2s_tx_fifo_enc
i2s_rx_fifo_uniq_1(i2s1_rx_fifo) - i2s_rx_fifo_enc

Figure 6.6. Setting Active Pins Constraints File

Functional simulation setup for Aldec Active-HDL are also included on each Ready to Use Configurations. The scripts can be executed from Active-HDL window through Lattice Radiant. The scripts can be found in the simulation/ice40up/ folder.

7. Resource Utilization

Resource utilization depends on the configurations. Table 7.1 shows the utilizations for the five Ready to Use Configurations.

Table 7.1. Resource Utilization

Configuration	Device	LUT	FF	EBR	PLL	I/O
Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8	M	1174	615	2	1	32
	S	1154	594	2	1	32
Configuration I2CMx6_GPIOx2	M	1700	947	0	1	20
	S	1945	1043	0	1	20
Configuration I2CMx1_GPIOx12	M	713	339	0	1	32
	S	740	349	0	1	23
Configuration I2CMx3_I2CSx2_GPIOx15	M	1578	813	0	1	31
	S	1526	798	0	1	31
Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8	M	524	227	1	1	9
	S	474	218	1	1	9

8. Single-Wire Aggregation Evaluation Board User Guide

8.1. SWA Evaluation Board Introduction

This iCE40 UltraPlus Single-Wire Aggregation (SWA) Evaluation Board is an easy-to-use platform for demonstrating and evaluating Ready-to-use SWA configurations. The board contains two iCE40 UltraPlus Devices. One of the devices (SWA FPGA) is used for the actual signal aggregation while the second device (Overhead FPGA) is used generating and verifying data like I^{2}C, I2S, and GPIO.

This guide describes how to begin using the SWA board. The contents of this user guide includes top-level functional descriptions of the various portions of the evaluation board, a summary of demonstrations, descriptions of the onboard connectors, switches, jumpers, configuration options, along with a complete set of schematics and the bill of materials.

Note: Static electricity can severely shorten the lifespan of electronic components. Be careful to follow proper ESD prevention handling standards when handling and using the iCE40 UltraPlus Single-Wire Aggregation (SWA) Evaluation Board.

8.2. Features

The iCE40 UltraPlus Single-Wire Aggregation (SWA) Evaluation Kit includes the items below.

- Two (2) Single-Wire Aggregation (SWA) Evaluation Board – The boards are used as the Master and Slave SWA boards.

  Key Components:

  - iCE40UP5K-48QFN (two pieces). One iCE40UP5K-48QFN acts the SWA FPGA while the other acts the Overhead FPGA. Overhead FPGA is added to generate and verify signals being aggregated and de-aggregated.
  - Power Regulation
  - I2S Microphones
  - Digital to Analog Converter audio amp with 3.5 mm connectors.
  - On-board connectors/pin header – To interconnect SWA and Overhead FPGA.
  - LED
  - Pushbutton Switches
- Pre-loaded Demo –a pre-loaded SWA demo included in the kit.
- Two USB Connector Cable – a mini USB port provides power, a programming interface.
- SWA and common ground wire – 1 meter wires to interconnect two SWA board.
- RJ45 connector – used as a single-wire link connection and also installed for future single-differential pair connection expansion.

Figure 8.1 shows the top side of the SWA Board indicating the specific features that are designed on the board.

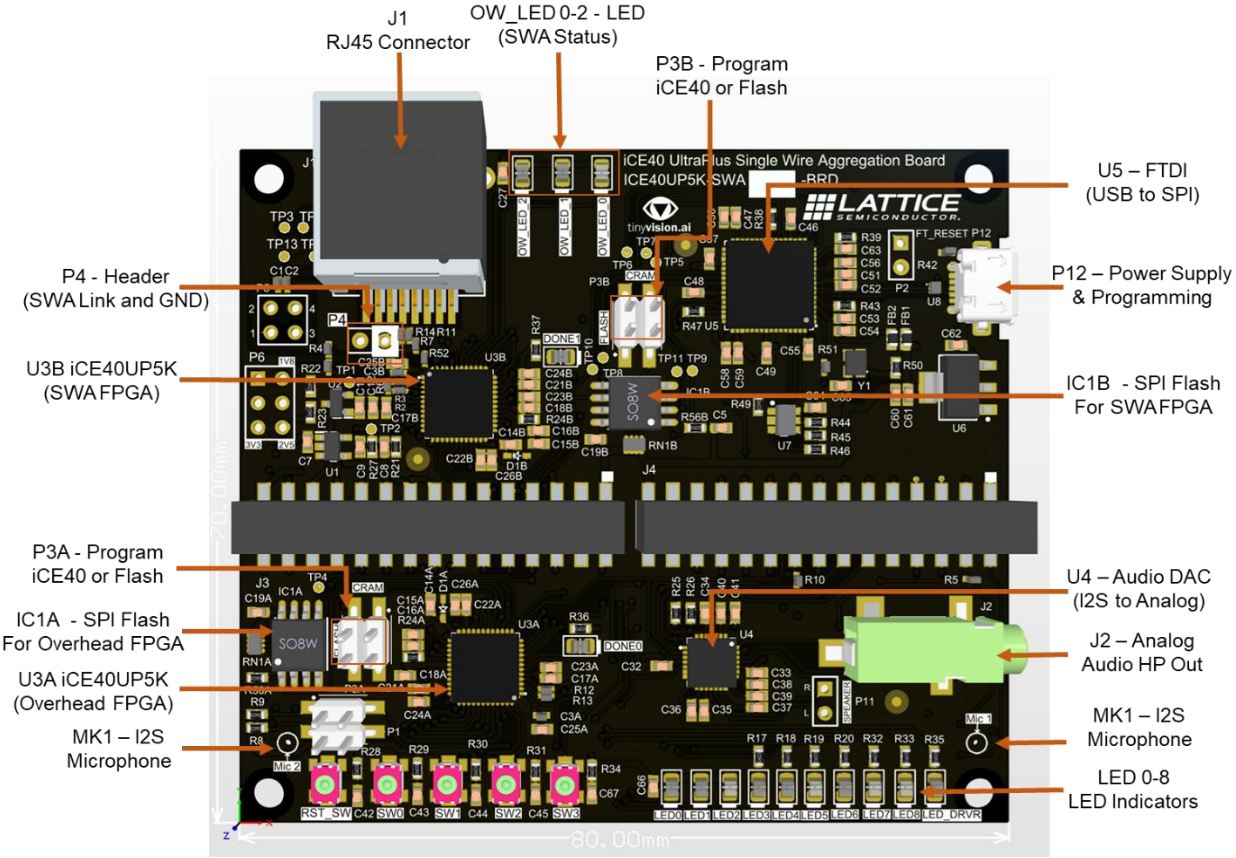


Figure 8.1. iCE40 UltraPlus Single-Wire Aggregation Evaluation Board (Top Side)

8.3. Clock Sources

A 12 MHz oscillator is built on the SWA board. This is used as a clock input for both SWA and Overhead FPGA. An external clock input can also be provided to the board. However, doing this requires removing resistor R51 and connecting the external clock source to port J3-28.

8.4. Software Requirements

Install the following software before evaluating the demo or ready to use configuration; or even developing designs for the board:

- Lattice Radiant Programmer 2.0 (or higher)
- Used for programming the iCE40 UltraPlus FPGA.
- Lattice Radiant Software 2.0 (or higher)
- Used for modifying/developing your own custom design for the iCE40 UltraPlus FPGA.

These software programs are available at the www.latticesemi.com/software. Make sure you log in to www.latticesemi.com, otherwise, these software downloads are not visible.

8.5. Board Configuration and Programming

The iCE40 UltraPlus Single-Wire Aggregation Evaluation Board has two iCE40 devices. iCE40 Device: U3B (SWA FPGA) is used for actual signal aggregation. While iCE40 Device: U3A (Overhead FPGA) is a complementary or optional FPGA. Overhead FPGA can be interconnected with SWA FPGA through Jumpers J3 and J4 to generate and verify signals being aggregated and de-aggregated.

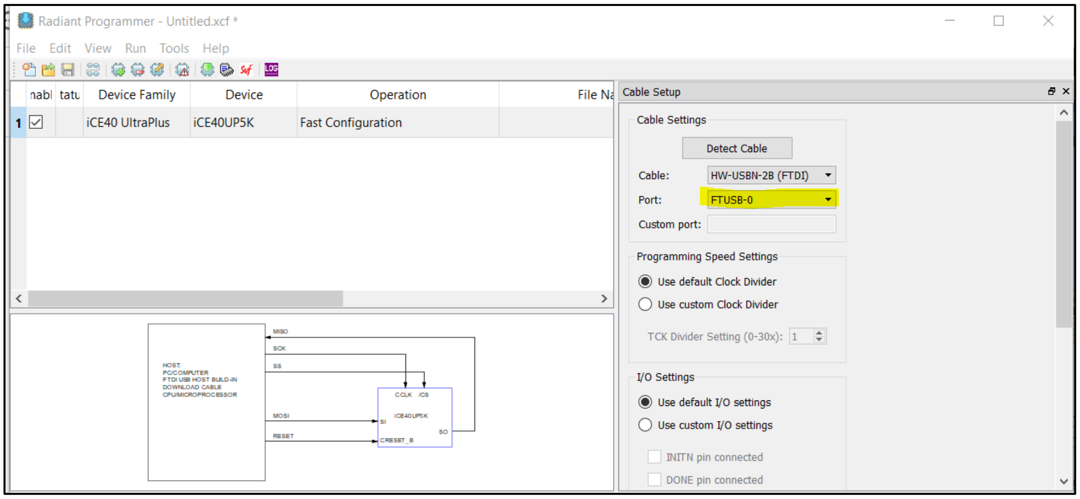
As there is only one USB to SPI bridge chip used onboard, programming either the SWA or Overhead FPGA devices can be completed using jumper setting at the board, and Port Setting at Lattice Radiant Programmer.

8.5.1. Jumpers Setting

- P3B – Short 1-2 and 3-4 to program external SPI FLASH for SWA FPGA. Short 1-3 and 2-4 to program FPGA Configuration Memory (CRAM) for SWA FPGA.
- P3A – Short 1-2 and 3-4 to program external SPI FLASH for Overhead FPGA. Short 1-3 and 2-4 to program FPGA CRAM for Overhead FPGA.

8.5.2. Lattice Radiant Programmer – Port Setting

- In Lattice Radiant Programmer under Cable Setting as shown in Figure 8.2. Choose the correct Port accordingly.
  - Choose FTUSB-1 to program SWA FPGA
  - Choose FTUSB-0 to program Overhead FPGA



Radiant Programmer - Untitled.xcf *

	nabl	tatu	Device Family	Device	Operation	File Na
1	☑		iCE40 UltraPlus	iCE40UP5K	Fast Configuration	

Cable Setup

Cable Settings

Cable: HW-USBN-2B (FTDI)

Port: FTUSB-0

Custom port:

Programming Speed Settings

● Use default Clock Divider

○ Use custom Clock Divider

TCK Divider Setting (0-30x): 1

I/O Settings

● Use default I/O settings

○ Use custom I/O settings

□ INITN pin connected

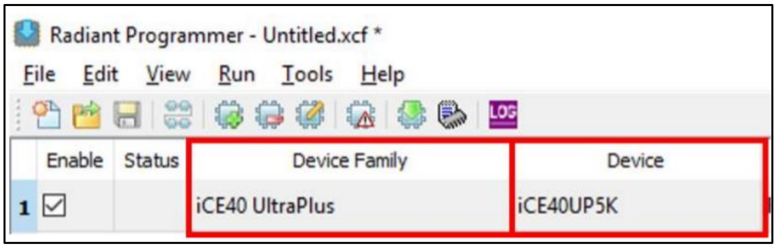
□ DONE pin connected

Figure 8.2. Port Setting at Lattice Radiant Programmer Software

8.5.3. Programming the SPI Flash

To program the SPI Flash

1. Short 1-2 and 3-4 of P3B (if SWA FPGA) or P3A (if Overhead FPGA).
2. Connect the SWA board via USB cable to PC with Lattice Radiant Programmer installed.
3. Start Lattice Radiant Programmer.
4. Set **Device Family** to **iCE40 UltraPlus** and **Device** to **iCE40UP5K** as shown in Figure 8.3.


Radiant Programmer - Untitled.xcf *
File Edit View Run Tools Help

	Enable	Status	Device Family	Device
1	☑		iCE40 UltraPlus	iCE40UP5K

Figure 8.3. Lattice Radiant Programmer: Device Family and Device

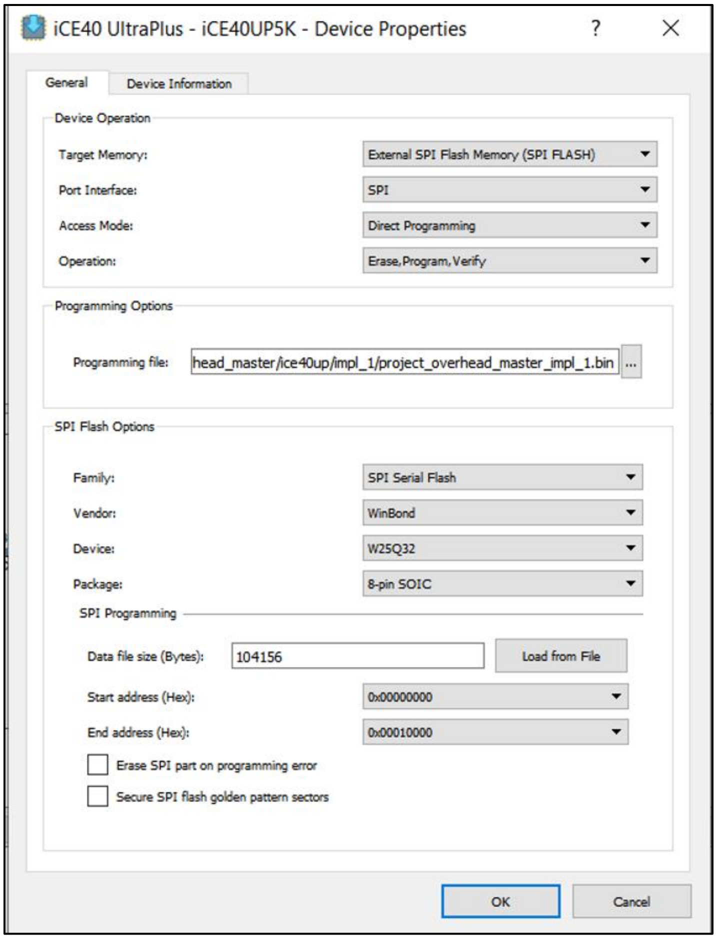
5. Click the iCE40 UltraPlus row and select **Edit > Device Properties.**
6. In the **Device Properties** dialog box, apply the settings below. See Figure 8.4.
    - Under **Device Operation**, select the options below:
    - Target Memory – External SPI Flash Memory (SPI FLASH)
    - Port Interface – SPI
    - Access Mode – Direct Programming
    - Operation – Erase, Program, Verify

    a. Under **Programming Options**, select the option below:

    - **Programming File** - <Select the desired file to program>

    b. Under **SPI Flash Options,** select the options below:

    - Family – SPI Serial Flash
    - Vendor – WinBond
    - Device – W25Q32
    - Package – 8-ipn SOIC
7. Click **OK** to close the **Device Properties** dialog box.
8. Select Cable Setting > Port.
9. Choose **FTUSB-1** to program SWA FPGA and **FTUSB-0** to program the Overhead FPGA.
10. Click the **Program** button in Lattice Radiant Programmer to program the onboard SPI Flash



iCE40 UltraPlus - iCE40UP5K - Device Properties

General | Device Information

Device Operation

Target Memory: External SPI Flash Memory (SPI FLASH)

Port Interface: SPI

Access Mode: Direct Programming

Operation: Erase,Program,Verify

Programming Options

Programming file: head_master/ice40up/impl_1/project_overhead_master_impl_1.bin

SPI Flash Options

Family: SPI Serial Flash

Vendor: WinBond

Device: W25Q32

Package: 8-pin SOIC

SPI Programming

Data file size (Bytes): 104156

Start address (Hex): 0x00000000

End address (Hex): 0x00010000

□ Erase SPI part on programming error

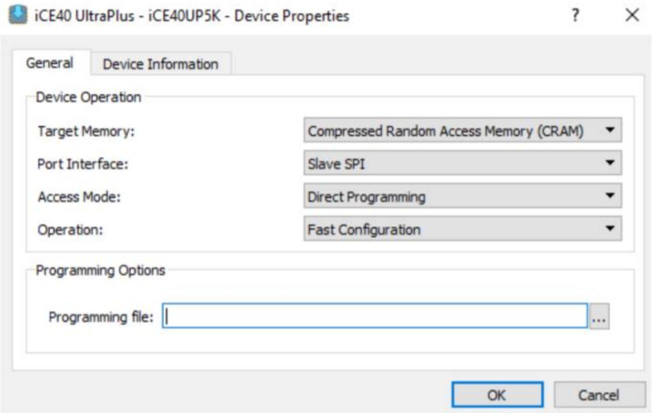
□ Secure SPI flash golden pattern sectors

Figure 8.4. Lattice Radiant Programmer: Device Family and Device Setting

8.5.4. Programming the CRAM Directly

To program the FPGA CRAM directly:

1. Short 1-3 and 2-4 of P3B (if SWA FPGA) or P3A (if Overhead FPGA)
2. Connect the SWA board via USB cable to PC with Lattice Radiant Programmer installed.
3. Start Lattice Radiant Programmer.
4. Set **Device Family** to **iCE40 UltraPlus** and **Device** to **iCE40UP5K** as shown in Figure 8.3.
5. Click the iCE40 UltraPlus row and select **Edit > Device Properties**.
6. In the **Device Properties** dialog box, apply the settings below. See Figure 8.5.
   - Under **Device Operation**, select the options below:
   - Target Memory – Compressed Random Access Memory (CRAM)
   - Port Interface – Slave SPI
   - Access Mode – Direct Programming
   - Operation – Fast Configuration
7. Click **OK** to close the **Device Properties** dialog box.
8. Select Cable Setting > Port.
9. Choose **FTUSB-1** to program SWA FPGA and **FTUSB-0** to program the Overhead FPGA.
10. Click the **Program** button in Lattice Radiant Programmer to program the onboard SPI Flash.


iCE40 UltraPlus - iCE40UP5K - Device Properties

General | Device Information

Device Operation

Target Memory: Compressed Random Access Memory (CRAM)

Port Interface: Slave SPI

Access Mode: Direct Programming

Operation: Fast Configuration

Programming Options

Programming file:

Figure 8.5. Lattice Radiant Programmer: Device Properties for iCE40 Device Configuration Memory

8.6. Demonstrations

This iCE40 UltraPlus Single-Wire Aggregation (SWA) Evaluation Board comes with a pre-configured SWA demo that demostrate aggregation of I2S, I^{2}C, and GPIO that demonstrate actual aggregation of I2S, I^{2}C, and GPIO signals. The SWA demo uses Overhead FPGA and other on-board components like I2S Microphone, DAC I2S to Analog converter, LEDs and switches to generate and verify signals being aggregated and de-aggregated on the Single-Wire link.

In addition, each ready-to-use configuration also has a corresponding configuration bitstream for the Overhead FPGA.

8.6.1. SWA Demo – Functional Overview

The functional block diagram shown in Figure 8.6 provides an overview of the SWA demo and how I^{2}S, I^{2}C and GPIO signals work through the Single-Wire Aggregation Evaluation board. This demo aggregates and de-aggregates the following signals.

- Channel 1: I2S audio signal from Master to Slave SWA Board.
  - Master SWA FPGA acts as the I2S controllers as it generates the I2S clock and ws for the I2S microphone and Master Overhead FPGA. I2S sampling rate is at ~48 kHz.
  - Master Overhead FPGA generate I2S data. I2S data being sent to Master SWA FPGA are either a 1 kHz single tone or I2S data coming from I2S microphone. Switching SW2 at Master SWA board selects the I2S data being sent to Master SWA FPGA.
  - De-aggregated I2S data from Slave SWA board are sent to Slave Overhead FPGA to verify if it is receiving a 1 kHz single tone. The same signals are sent to DAC so that you can verify the received audio through the audio jack. LED0 on Slave SWA FPGA indicates the status of I2S data verification. Blinking LED0 means its receiving expected single tone data.
  - Switching SW2 at the Slave SWA board resets I2S verification being done by Slave Overhead FPGA.
- Channel 2: I^{2}C signal from Master to Slave Board.
  - Master Overhead FPGA acts as I^{2}C Master and generates I^{2}C commands to enable, turn-on/mute DAC.
  - De-aggregated I^{2}C data from Slave SWA are sent to DAC, which act as the slave I^{2}C.
  - Switching SW0 at Master SWA Board initiate Master Overhead FPGA to generate the I^{2}C commands.
- Channel 3: 6-bit GPIO Signal Master to Slave Board. 6-bit GPIO Signal Slave to Master Board.
  - Overhead FPGA generates the 6 bit counter GPIO Data that is being sent to the SWA FPGA. Receiving SWA FPGA de-aggregates the GPIO data. The data are then sent it to LED[8:3] for visual verification and Overhead FPGA to verify if it is receiving 6 bit counter GPIO data. LED0 on the receiving SWA Board indicates the status of GPIO data verification. Blinking LED0 means its receiving expected 6 bit counter data.
  - Switching SW1 on the transmitting SWA board resets the 6-bit counter data.
  - Switching SW1 on the receiving SWA board resets the GPIO verification.

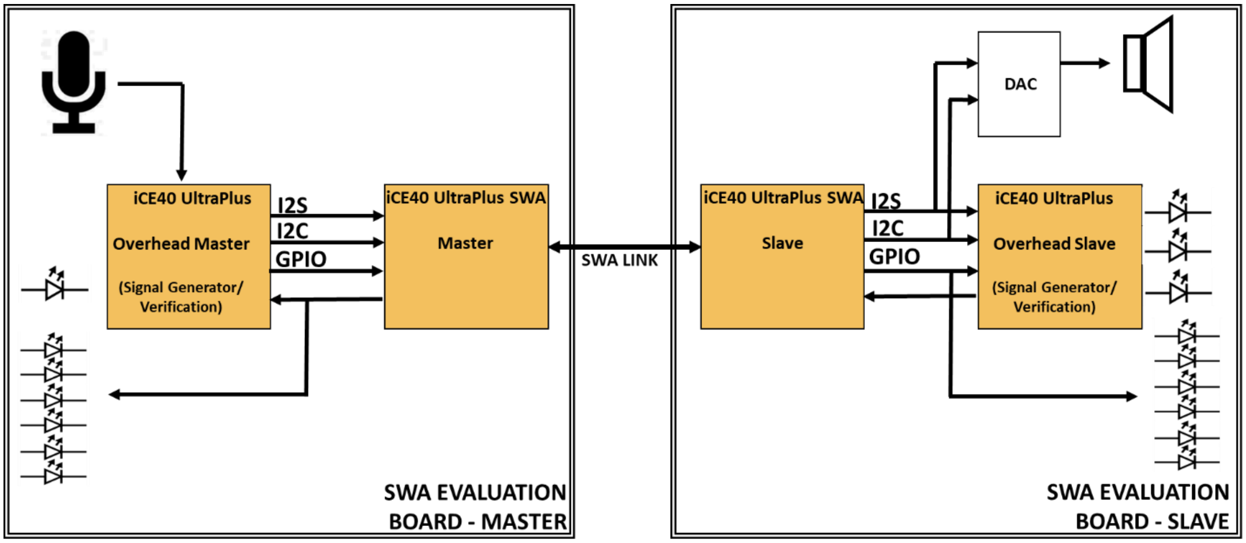


Figure 8.6. Functional Block Diagram

8.6.2. Setting Up SWA Demo

To set up the SWA demo:

1. Ensure Jumper Connection on the SWA Board as shown in Figure 8.7.
   - Port J3: Short 1-2, 3-4, 5-6, 7-8, 9-10, 11-12, 13-14, 15-16, 17-18, 19-20, 21-22, 23-24, 25-26, 27-28, 29-30. This interconnect IO of SWA and Overhead FPGA.
   - Port J4: Short 1-2, 3-4, 5-6, 7-8, 9-10, 11-12, 13-14, 15-16, 17-18, 19-20, 21-22, 23-24, 25-26, 27-28, 29-30. This interconnect IO of SWA and Overhead FPGA.
   - Port P3A: Short 1-2 and 3-4. This is to program SPI Flash for Overhead FPGA.
   - Port P3B: Short 1-2 and 3-4. This is to program SPI Flash for SWA FPGA.
   - Port P5: Short 1-2 and 3-4. This interconnect Pin 44 of SWA FPGA to Pin 3 of Overhead FPGA; Pin 47 of SWA FPGA to Pin 4 of Overhead FPGA.
   - Port P1: Short 1-2, and 3-4. This put 4.7 k Ω pull-up resistor on I^{2}C port.
   - Port P8: Short 1-2. This set VCCIO bank 2 to 3.3 volts.

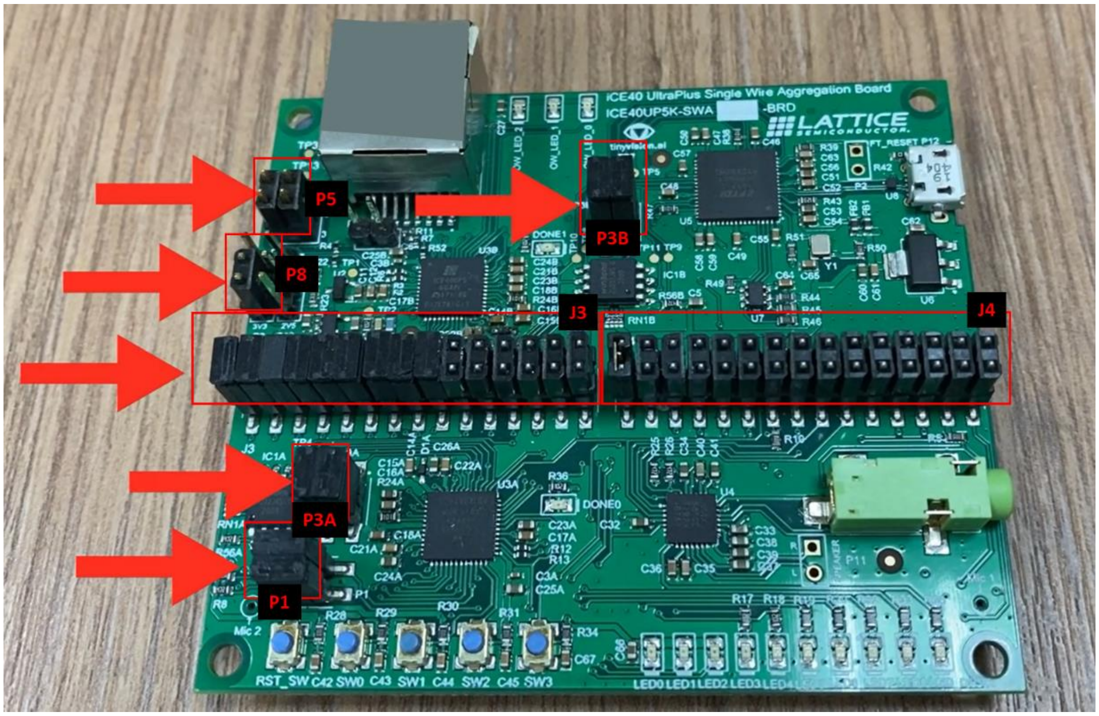
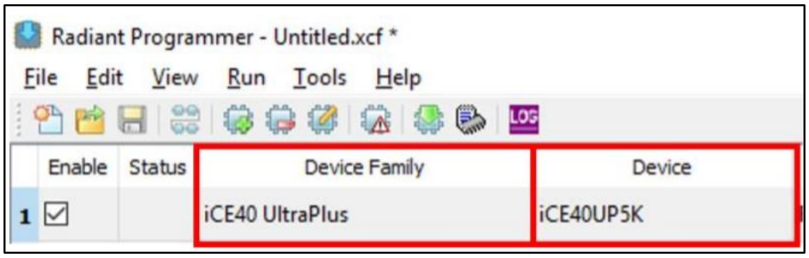


Figure 8.7. SWA Demo Port Connection

2. Program SPI Flash for Master SWA FPGA – Master SWA Board.
    a. Connect the SWA board via USB cable to PC with Lattice Radiant Programmer installed.
    b. Start Lattice Radiant Programmer.
    c. Set **Device Family** to **iCE40 UltraPlus** and **Device** to **iCE40UP5K** as shown in Figure 8.8.



	Enable	Status	Device Family	Device
1	☑		iCE40 UltraPlus	iCE40UP5K

Figure 8.8. Lattice Radiant Programmer: Device Family and Device Setting

    d. Click the iCE40 UltraPlus row and select **Edit > Device Properties.**
    e. In the **Device Properties** dialog box, apply the settings below.
    f. Under **Device Operation**, select the options below:

- Target Memory – External SPI Flash Memory (SPI FLASH)
- Port Interface – SPI
- Access Mode – Direct Programming
- Operation – Erase, Program, Verify

    g. Under **Programming Options**, select the option below:

- **Programming File – ConfigurationDemo_Master.bin** (Bitstream can be found on the SWA projects under bitstream/demo)

    h. Under **SPI Flash Options,** select the options below:

- Family – SPI Serial Flash
- Vendor – WinBond
- Device – W25Q32
- Package – 8-ipn SOIC

i. Click **OK** to close the **Device Properties** dialog box.

j. Select Cable Setting > Port.

k. Choose **FTUSB-1** to program SWA FPGA and **FTUSB-0** to program Overhead FPGA.

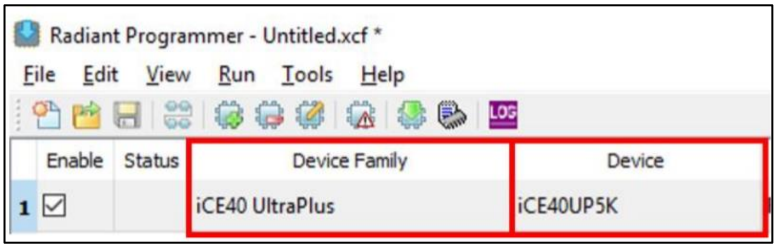
l. Click the **Program** button in Lattice Radiant Programmer to program the onboard SPI Flash.

3. Program SPI Flash for Master Overhead FPGA – Master SWA Board.

a. Connect the SWA board via USB cable to PC with Lattice Radiant Programmer installed.

b. Start Lattice Radiant Programmer.

c. Set **Device Family** to **iCE40 UltraPlus** and **Device** to **iCE40UP5K** as shown in Figure 8.9.



	Enable	Status	Device Family	Device
1	☑		iCE40 UltraPlus	iCE40UP5K

Figure 8.9. Lattice Radiant Programmer: Device Family and Device Setting

d. Click the iCE40 UltraPlus row and select **Edit > Device Properties.**

e. In the **Device Properties** dialog box, apply the settings below.

f. Under **Device Operation**, select the options below:

- Target Memory – External SPI Flash Memory (SPI FLASH)
- Port Interface – SPI
- Access Mode – Direct Programming
- Operation – Erase, Program, Verify

g. Under **Programming Options**, select the option below:

- **Programming File – Overhead_ConfigurationDemo_Master.bin** (Bitstream can be found on the SWA projects under bitstream/demo)

h. Under **SPI Flash Options,** select the options below:

- Family – SPI Serial Flash
- Vendor – WinBond
- Device – W25Q32
- Package – 8-ipn SOIC

i. Click **OK** to close the **Device Properties** dialog box.

j. Select Cable Setting > Port.

k. Choose **FTUSB-0**.

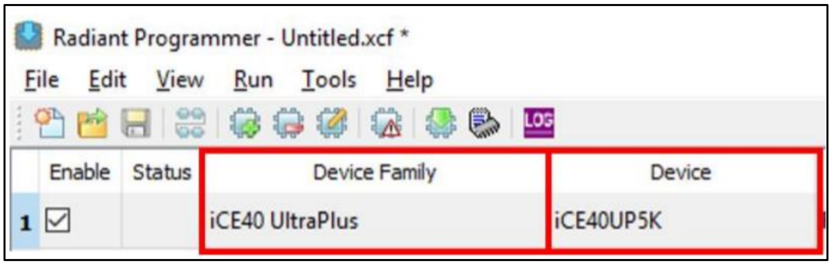
l. Click the **Program** button in Lattice Radiant Programmer to program the onboard SPI Flash.

4. Program SPI Flash for Slave SWA FPGA – Slave SWA Board.

a. Connect the SWA board through USB cable to the PC with Lattice Radiant Programmer installed.

b. Start Lattice Radiant Programmer.

c. Set **Device Family** to **iCE40 UltraPlus** and **Device** to **iCE40UP5K** as shown in Figure 8.10.


Radiant Programmer - Untitled.xcf *
File Edit View Run Tools Help
LOG

	Enable	Status	Device Family	Device
1	☑		iCE40 UltraPlus	iCE40UP5K

Figure 8.10. Lattice Radiant Programmer: Device Family and Device Setting

d. Click the iCE40 UltraPlus row and select **Edit > Device Properties.**

e. In the **Device Properties** dialog box, apply the settings below.

f. Under **Device Operation**, select the options below:

- Target Memory – External SPI Flash Memory (SPI FLASH)
- Port Interface – SPI
- Access Mode – Direct Programming
- Operation – Erase, Program, Verify

g. Under **Programming Options**, select the option below:

- **Programming File – ConfigurationDemo_Slave.bin** (Bitstream can be found on the SWA projects under bitstream/demo)

h. Under **SPI Flash Options,** select the options below:

- Family – SPI Serial Flash
- Vendor – WinBond
- Device – W25Q32
- Package – 8-ipn SOIC

i. Click **OK** to close the **Device Properties** dialog box.

j. Select Cable Setting > Port.

k. Choose **FTUSB-1** to program SWA FPGA and **FTUSB-0** to program the Overhead FPGA.

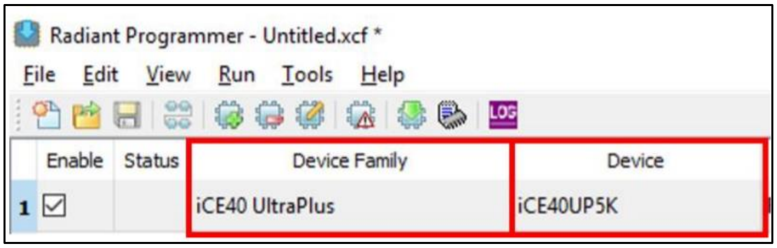
l. Click the **Program** button in Lattice Radiant Programmer to program the onboard SPI Flash.

5. Program SPI Flash for Slave Overhead FPGA – Slave SWA Board.

a. Connect the SWA board via USB cable to PC with Lattice Radiant Programmer installed.

b. Start Lattice Radiant Programmer.

c. Set **Device Family** to **iCE40 UltraPlus** and **Device** to **iCE40UP5K** as shown in Figure 8.11.


Radiant Programmer - Untitled.xcf *
File Edit View Run Tools Help
LOG

	Enable	Status	Device Family	Device
1	☑		iCE40 UltraPlus	iCE40UP5K

Figure 8.11. Lattice Radiant Programmer: Device Family and Device Setting

d. Click the iCE40 UltraPlus row and select **Edit > Device Properties**.

e. In the **Device Properties** dialog box, apply the settings below.

f. Under **Device Operation**, select the options below:

- Target Memory – External SPI Flash Memory (SPI FLASH)
- Port Interface – SPI
- Access Mode – Direct Programming
- Operation – Erase, Program, Verify

g. Under **Programming Options**, select the option below:

- **Programming File – Overhead_ConfigurationDemo_Slave.bin** (Bitstream can be found on the SWA projects under bitstream/demo)

h. Under **SPI Flash Options,** select the options below:

- Family – SPI Serial Flash
- Vendor – WinBond
- Device – W25Q32
- Package – 8-ipn SOIC

i. Click **OK** to close the **Device Properties** dialog box.

j. Select Cable Setting > Port.

k. Choose **FTUSB-0** to program **SWA FPGA** and **FTUSB-0** to program the **Overhead FPGA**.

l. k. Click the **Program** button in Lattice Radiant Programmer to program the onboard SPI Flash.

6. Connect the Single-Wire Link and common ground on Port P4 as shown in Figure 8.12.

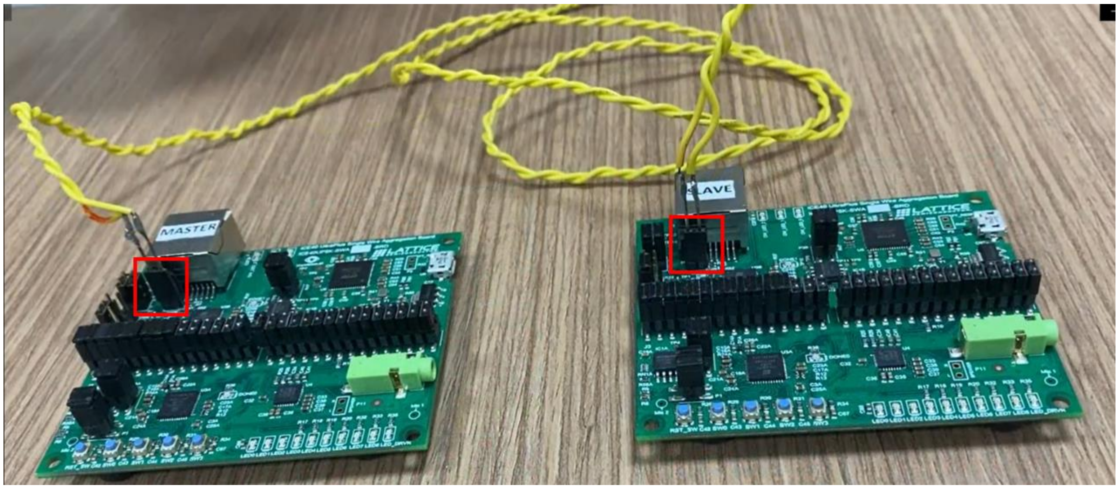


Figure 8.12. Connecting Single-Wire Link and Common Ground at Port P4.

7. Connect a headphone on the Slave SWA board Audio Jack (J2).
8. Power-up the SWA board by connecting USB power cable as shown in Figure 8.13.

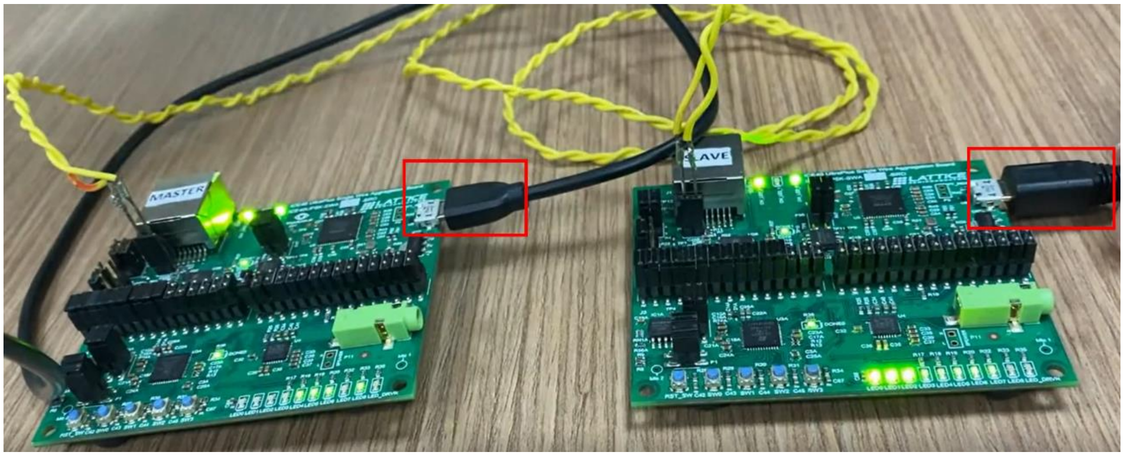


Figure 8.13. Connecting USB power cable.

8.6.3. Running the SWA Demo

This section discusses how to run the Single-wire Signal Aggregation demo. For more information, a training video on how the SWA demo work is also available for reference.

To run the SWA demo:

1. Press and hold the respective RST_SW buttons of both Master and Slave SWA board.
2. Release the button of each SWA board. This resets the SWA FPGA.
3. Simultaneously press SW3 of each board. This resets the Overhead FPGA.
4. Note the following after resetting Overhead FPGA.
   - GPIO Signal Aggregation
   - Each Overhead FPGA (on Master and Slave SWA Board) starts sending 6-bit counter GPIO signals to the SWA FPGA for aggregation. It is then de-aggregated by the receiving SWA FPGA. After which, the de-aggregated signals are sent to the receiving Overhead FPGA and LED for verification.
   - On both SWA Board, LED[8:3] blink like a 6-bit counter.
   - On both SWA Board, LED2 blinks. This indicates that the receiving SWA board is receiving 6 bit counter GPIO signals.
   - Pressing SW1 resets GPIO generation and verification. Pressing SW1 on the transmitting SWA board causes LED2 of the receiving board turn off because it interrupts the expected 6-bit counter GPIO signals. Pressing SW1 on the receiving SWA board resets the GPIO verification. After this, the LED2 blinks again.
   - I^{2}C Signal Aggregation
   - At Master SWA Board, Press SW0. This causes the Master Overhead FPGA to generate 9 I^{2}C commands to set-up, enable DAC on the Slave SWA Board.
   - 1 kHz single tone or audio coming from the I2S microphone of the Master SWA board is observed on the audio receiver connected on the Slave SWA board's audio jack.
   - Pressing SW0 again generates I^{2}C command to mute and unmute the DAC.
   - I2S Signal Aggregation
   - Master SWA FPGA acts as the I2S controllers as it generates the I2S clock and I2S WS for the I2S microphone and Master Overhead FPGA. I2S sampling rate is at ~48 kHz.
   - Master Overhead FPGA generate I2S data. I2S data being sent to Master SWA FPGA are either a 1 kHz single tone or I2S data coming from I2S microphone. Pressing or switching SW2 at Master SWA board selects which I2S data being sent to Master SWA FPGA.
   - De-aggregated I2S data from Slave SWA board are sent to Slave Overhead FPGA to verify if it is receiving a 1 kHz single tone. Same signal are sent to DAC so that you can verify the received audio through audio jack. LED0

on Slave SWA FPGA indicates the status of I2S data verification. Blinking LED0 means its receiving expected single tone data.

- Switching SW2 at Slave SWA board resets I2S verification being done by Slave Overhead FPGA.
- I2S Signals are also feedback to Master Overhead FPGA. After resetting Master Overhead FPGA, Master SWA LED0 is blinking which indicate it is sending 1 kHz single tone signal to SWA FPGA.
- Switching Slave SWA Board – SW2 resets I2S verification.

8.6.4. Evaluation Demo For: Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8

Aside from the SWA demo, each ready-to-use configuration has a configuration bitstream for Master and Slave Overhead FPGA for evaluation purposes. The functional block diagram shown in Figure 8.14 provides an overview of the SWA Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8 demo and how I2S, I^{2}C, and GPIO signals work through the Single-Wire Aggregation Evaluation board. This demo aggregates and de-aggregates the following signals.

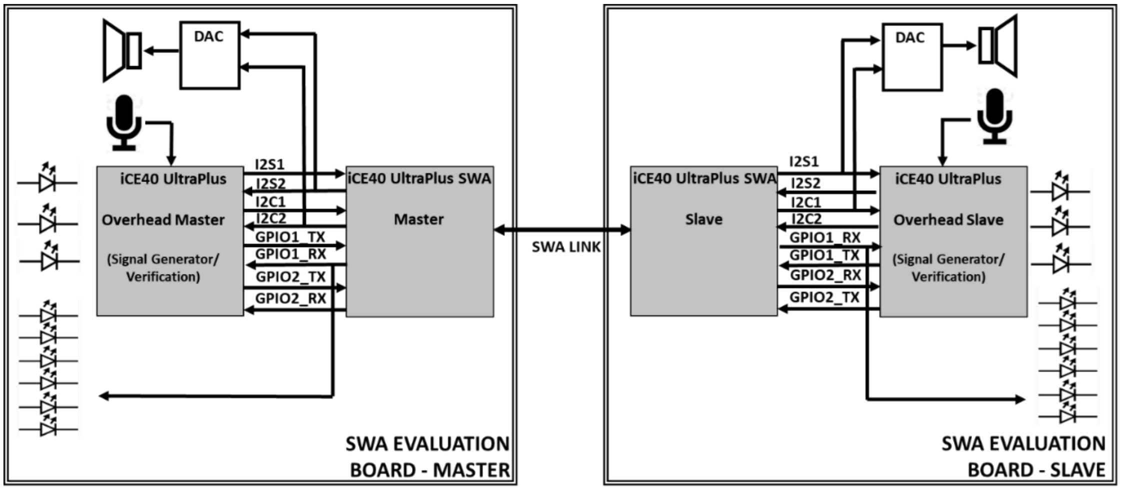


Figure 8.14. Functional Block Diagram for Configuration I2Sx2_I2CSx1_I2CMx1_GPIOx8 Evaluation

- GPIO Signal Aggregation:
- Each Overhead FPGA (on Master and Slave SWA Board) generate 6-bit counter GPIO signals for GPIO1, and 2-bit counter signals for GPO2, which eventually send to SWA FPGA for aggregation. It is then de-aggregated by the receiving SWA FPGA. The de-aggregated signals are sent to the receiving Overhead FPGA for verification. Received GPIO1 signals are send to LED[8:3] for visual verification.
- On both SWA Board, LED[8:3] blinks like a 6-bit counter.
- On both SWA Board, LED2 blinks. This indicates that the receiving SWA board is receiving 6-bit counter GPIO signals on GPIO1 and 2-bit counter GPIO signals on GPIO2.
- Pressing SW1 resets GPIO generation and verification. Pressing SW1 on the transmitting SWA board causes LED2 of the receiving board to turn off because it interrupts the expected GPIO signals (Both GPIO1 and GPIO2). Pressing SW1 on the receiving SWA board resets the GPIO verification. After this, the LED2 blinks again.
- I^{2}C Signal Aggregation:
- For I2C1:
- At Master SWA Board, Press SW0. This causes the Master Overhead FPGA to generate 9 I^{2}C commands to set-up, enable DAC on the Slave SWA Board.
- 1 kHz single tone or audio coming from the I2S microphone of the Master SWA board is observed on the audio receiver connected on the Slave SWA board’s audio jack.

- Pressing Master SWA Board - SW0 again generates I^{2}C command to mute and unmute the Slave SWA Board - DAC.
- For I2C2:
- At Slave SWA Board, Press SW0. This causes the Slave Overhead FPGA to generate 9 I^{2}C commands to set-up, enable DAC on the Master SWA Board.
- 1 kHz single tone or audio coming from the I2S microphone of the Slave SWA board is observed on the audio receiver connected on the Master SWA board's audio jack.
- Pressing Slave SWA Board - SW0 again generates I^{2}C command to mute and unmute the Master SWA Board - DAC.
- I2S Signal Aggregation:
- For I2S1:
- Master SWA FPGA acts as the I2S controllers as it generates the I2S clock and I2S WS for the I2S microphone and Master Overhead FPGA. I2S sampling rate is at ~32 kHz.
- Master Overhead FPGA generate I2S data. I2S data being sent to Master SWA FPGA are either a 1 kHz single tone or I2S data coming from Master SWA Board - I2S microphone. Pressing or switching SW2 at Master SWA board selects the I2S data being sent to Master SWA FPGA.
- De-aggregated I2S data from Slave SWA board are sent to Slave Overhead FPGA to verify if it is receiving a 1 kHz single tone. Same signal are sent to DAC so that you can verify the received audio through audio jack. LED0 on Slave SWA FPGA indicates the status of I2S data verification. Blinking LED0 means its receiving expected single tone data.
- Switching SW2 at Slave SWA board resets I2S verification being done by Slave Overhead FPGA.
- Switching Slave SWA Board – SW2 resets I2S verification.
- For I2S2:
- Slave SWA FPGA Aggregate I2S Channel 2 to Master SWA FPGA. It acts as a I2S slave. I2S data, I2S clock and I2S WS are being generated by Slave Overhead FPGA that is send to Slave FPGA for aggregation. I2S sampling rate is at ~32 kHz.
- I2S data being sent to Slave SWA FPGA are either a 1 kHz single tone or I2S data coming from Slave SWA Board - I2S microphone. Pressing or switching SW2 at Slave SWA board selects the I2S data being sent to Slaves WA FPGA.
- De-aggregated I2S data from Master SWA board are sent to Master Overhead FPGA to verify if it is receiving a 1 kHz single tone. Same signal are sent to Master SWA Board - DAC so that you can verify the received audio through audio jack. LED0 on Master SWA FPGA indicates the status of I2S data verification. Blinking LED0 means its receiving expected single tone data.
- Switching SW2 at Master SWA board resets I2S verification being done by Master Overhead FPGA.
- Switching Master SWA Board – SW2 resets I2S verification.

8.6.5. Evaluation Demo For: Configuration I2CMx6_GPIOx6

Aside from SWA Demo, each ready-to-use configuration have a configuration bitstream for Master and Slave Overhead FPGA for evaluation purposes. The functional block diagram shown in Figure 8.15 provides an overview of the SWA Configuration I2CMx6_GPIOx6 Evaluation demo and how I^{2}C and GPIO signals work through the Single-Wire Aggregation Evaluation board. This demo aggregates and de-aggregates following signals.

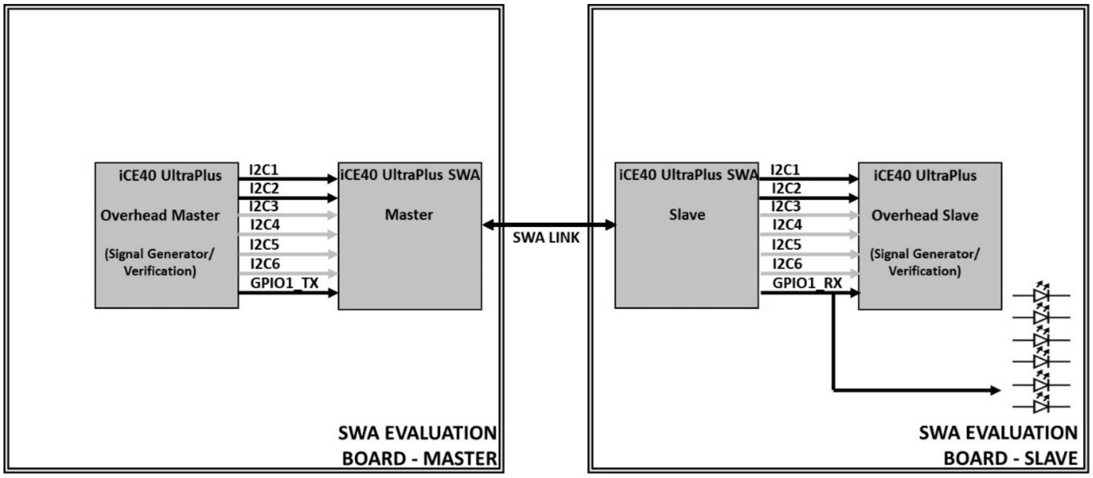


Figure 8.15. Functional Block Diagram for Configuration I2CMx6_GPIOx6 Evaluation

- GPIO Signal Aggregation:
- Master Overhead FPGA generate 6-bit counter GPIO signals for GPIO1, which eventually send to Master SWA FPGA for aggregation. It is then de-aggregated by the Slave SWA FPGA. The de-aggregated signals are sent to the Slave Overhead FPGA for verification. Received GPIO1 signals are send to LED[8:3] for visual verification.
- On Slave SWA Board, LED[8:3] blinks like a 6-bit counter.
- On Slave SWA Board, LED2 blinks. This indicates that the receiving SWA board is receiving 6-bit counter GPIO signals on GPIO1.
- Pressing Master SWA Board - SW1 resets GPIO generation. Pressing Slave SWA Board – SW1 resets GPIO verification. Pressing SW1 on the Master SWA board causes LED2 of the Slave SWA Board turn off because it interrupts the expected GPIO signals. Pressing SW1 on the Slave SWA board resets the GPIO verification. After this, the LED2 blinks again.
- I^{2}C Signal Aggregation:
- For I2C1 and I2C2:
- At Master SWA Board, Press SW0. This causes the Master Overhead FPGA to generate I^{2}C commands for I^{2}C Channel 1 and 2.
- At Slave SWA Board, de-aggregated I^{2}C signals are received by Slave Overhead FPGA as an I^{2}C slave. Slave SWA Board – LED1 turns-On when it is receiving a valid and expected I^{2}C Command.
- Overhead FPGA did not generate an I^{2}C signals for I2C3 – I2C6. I^{2}C signals from I2C1 and I2C2 can be use using jumpers at Port J3 and J4.
- You may use I^{2}C Channel: I2C3-I2C6 to connect their own component for evaluation.

8.6.6. Evaluation Demo For: Configuration I2CMx1_GPIOx12

Aside from SWA Demo, each ready-to-use configuration have a configuration bitstream for Master and Slave Overhead FPGA for evaluation purposes. The functional block diagram shown in Figure 8.16 provides an overview of the SWA Configuration I2CMx1_GPIOx12 Evaluation demo and how I^{2}C and GPIO signals work through the Single-Wire Aggregation Evaluation board. This demo aggregates and de-aggregates the following signals.

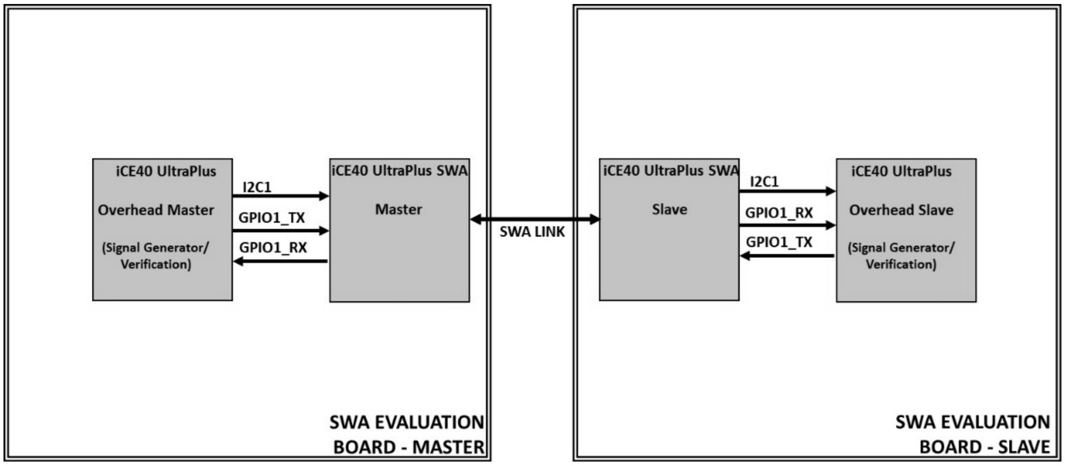


Figure 8.16. Functional Block Diagram for Configuration I2CMx1_GPIOx12 Evaluation

- GPIO Signal Aggregation:
  - Each Overhead FPGA (on Master and Slave SWA Board) generate 12-bit counter GPIO signals for GPIO1, which eventually send to SWA FPGA for aggregation. It is then de-aggregated by the receiving SWA FPGA. The de-aggregated signals are sent to the receiving Overhead FPGA for verification. Received GPIO1 bit 0-5 signals are send to LED[8:3] for visual verification.
  - On both SWA Board, LED[8:3] blinks like a 6-bit counter.
  - On both SWA Board, LED2 blinks. This indicates that the receiving SWA board is receiving 12-bit counter GPIO signals on GPIO1.
  - Pressing SW1 resets GPIO generation and verification. Pressing SW1 on the transmitting SWA board causes LED2 of the receiving board turn off because it interrupts the expected GPIO signals. Pressing SW1 on the receiving SWA board resets the GPIO verification. After this, the LED2 blinks again.
- I^{2}C Signal Aggregation:
  - Press SW2 on both Board to reset the Overhead FPGA.
  - At Master SWA Board, press SW0. This causes the Master Overhead FPGA to generate I^{2}C commands for I2C1.
  - At Slave SWA Board, de-aggregated I^{2}C signals are received by Slave Overhead FPGA as an I^{2}C slave. Slave SWA Board – LED1 turns-On when it is receiving a valid and expected I^{2}C Command.

8.6.7. Evaluation Demo For: Configuration I2CMx3_I2CSx2_GPIOx15

Aside from SWA Demo, each ready-to-use configuration have a configuration bitstream for Master and Slave Overhead FPGA for evaluation purposes. The functional block diagram shown in Figure 8.17 provides an overview of the SWA Configuration I2CMx3_I2Cx2_GPIOx15 Evaluation demo and how I^{2}C and GPIO signals work through the Single-Wire Aggregation Evaluation board. This demo aggregates and de-aggregates following signals.

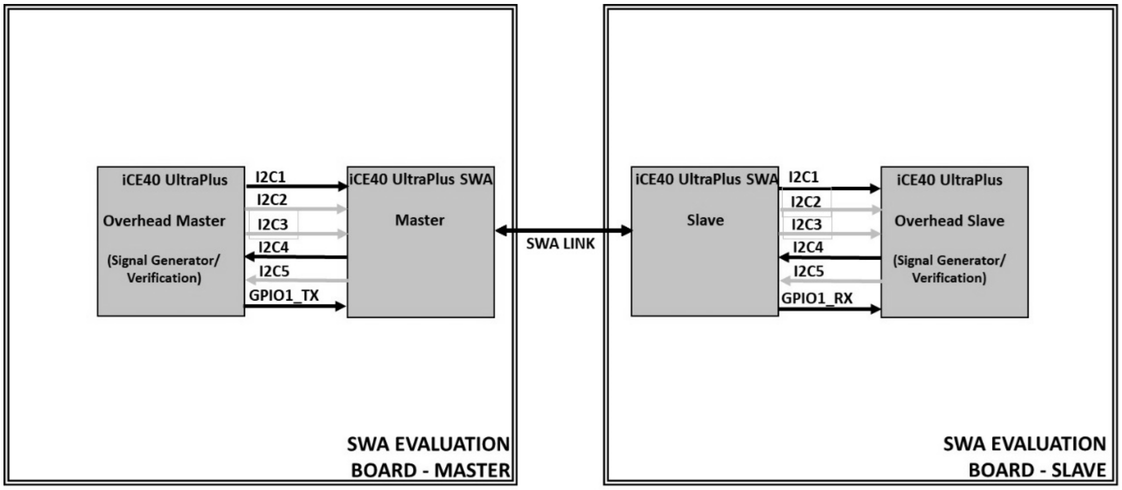


Figure 8.17. Functional Block Diagram for Configuration I2CMx3_I2CSx2_GPIOx15 Evaluation

- GPIO Signal Aggregation:
  - At Port P1, keep the pin header open.
  - Press SW2 of each SWA board to Reset Overhead FPGA
  - Master Overhead FPGA generate 15-bit counter GPIO signals for GPIO1, which eventually send to Master SWA FPGA for aggregation. It is then de-aggregated by the Slave SWA FPGA. The de-aggregated signals are sent to the Slave Overhead FPGA for verification. Received GPIO1 [5:0] signals are send to LED[8:3] for visual verification.
  - On Slave SWA Board, LED[8:3] blinks like a 6-bit counter.
  - On Slave SWA Board, LED2 blinks. This indicates that the receiving SWA board is receiving 15-bit counter GPIO signals on GPIO1.
  - Pressing Master SWA Board - SW1 resets GPIO generation. Pressing Slave SWA Board – SW1 resets GPIO verification. Pressing SW1 on the Master SWA board causes LED2 of the Slave SWA Board turn off because it interrupts the expected GPIO signals. Pressing SW1 on the Slave SWA board resets the GPIO verification. After this, the LED2 blinks again.
- I^{2}C Signal Aggregation:

  For I2C1:
  - At Master SWA Board, Press SW0. This causes the Master Overhead FPGA to generate I^{2}C commands for I^{2}C Channel 1.
  - At Slave SWA Board, de-aggregated I^{2}C signals are received by Slave Overhead FPGA as an I^{2}C slave. Slave SWA Board – LED1 turns-On when it is receiving a valid and expected I^{2}C Command.

  For I2C4:
  - At Slave SWA Board, Press SW0. This causes the Slave Overhead FPGA to generate I^{2}C commands for I^{2}C Channel 4.
  - At Master SWA Board, de-aggregated I^{2}C signals are received by Master Overhead FPGA as an I^{2}C slave. Master SWA Board – LED1 turns-On when it is receiving a valid and expected I^{2}C Command.
  - You may use I^{2}C Channel: I2C2, I2C3, I2C5 to connect their own component for evaluation.

8.6.8. Evaluation Demo For: Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8

Aside from SWA Demo, each ready-to-use configuration have a configuration bitstream for Master and Slave Overhead FPGA for evaluation purposes. The functional block diagram shown in Figure 8.18 provides an overview of the SWA Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8 demo and how I^{2}S, I^{2}C, and GPIO signals work through the Single-Wire Aggregation Evaluation board. This demo aggregates and de-aggregates following signals.

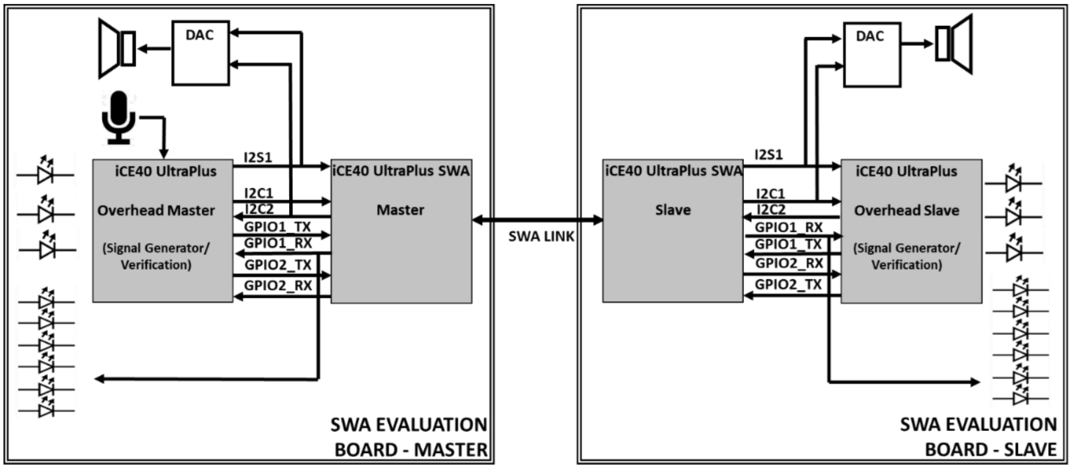


Figure 8.18. Functional Block Diagram for Configuration I2Sx1_I2CMx1_I2CSx1_GPIOx8 Evaluation

- GPIO Signal Aggregation:
  - Each Overhead FPGA (on Master and Slave SWA Board) generate 6-bit counter GPIO signals for GPIO1, and 2-bit counter signals for GPO2, which eventually send to SWA FPGA for aggregation. It is then de-aggregated by the receiving SWA FPGA. The de-aggregated signals are sent to the receiving Overhead FPGA for verification. Received GPIO1 signals are send to LED[8:3] for visual verification.
  - On both SWA Board, LED[8:3] blinks like a 6-bit counter.
  - On both SWA Board, LED2 blinks. This indicates that the receiving SWA board is receiving 6-bit counter GPIO signals on GPIO1 and 2-bit counter GPIO signals on GPIO2.
  - Pressing SW1 resets GPIO generation and verification. Pressing SW1 on the transmitting SWA board causes LED2 of the receiving board turn off because it inerrupts the expected GPIO signals (Both GPIO1 and GPIO2). Pressing SW1 on the receiving SWA board resets the GPIO verification. After this, the LED2 blinks again.
- I2C Signal Aggregation:

  For I2C1 :
  - At Master SWA Board, Press SW0. This causes the Master Overhead FPGA to generate 9 I^{2}C commands to set-up, enable DAC on the Slave SWA Board.
  - 1 kHz single tone or audio coming from the I2S microphone of the Master SWA board is observed on the audio receiver connected on the Slave SWA board’s audio jack.
  - Pressing Master SWA Board - SW0 again generates I^{2}C command to mute and unmute the Slave SWA Board - DAC.

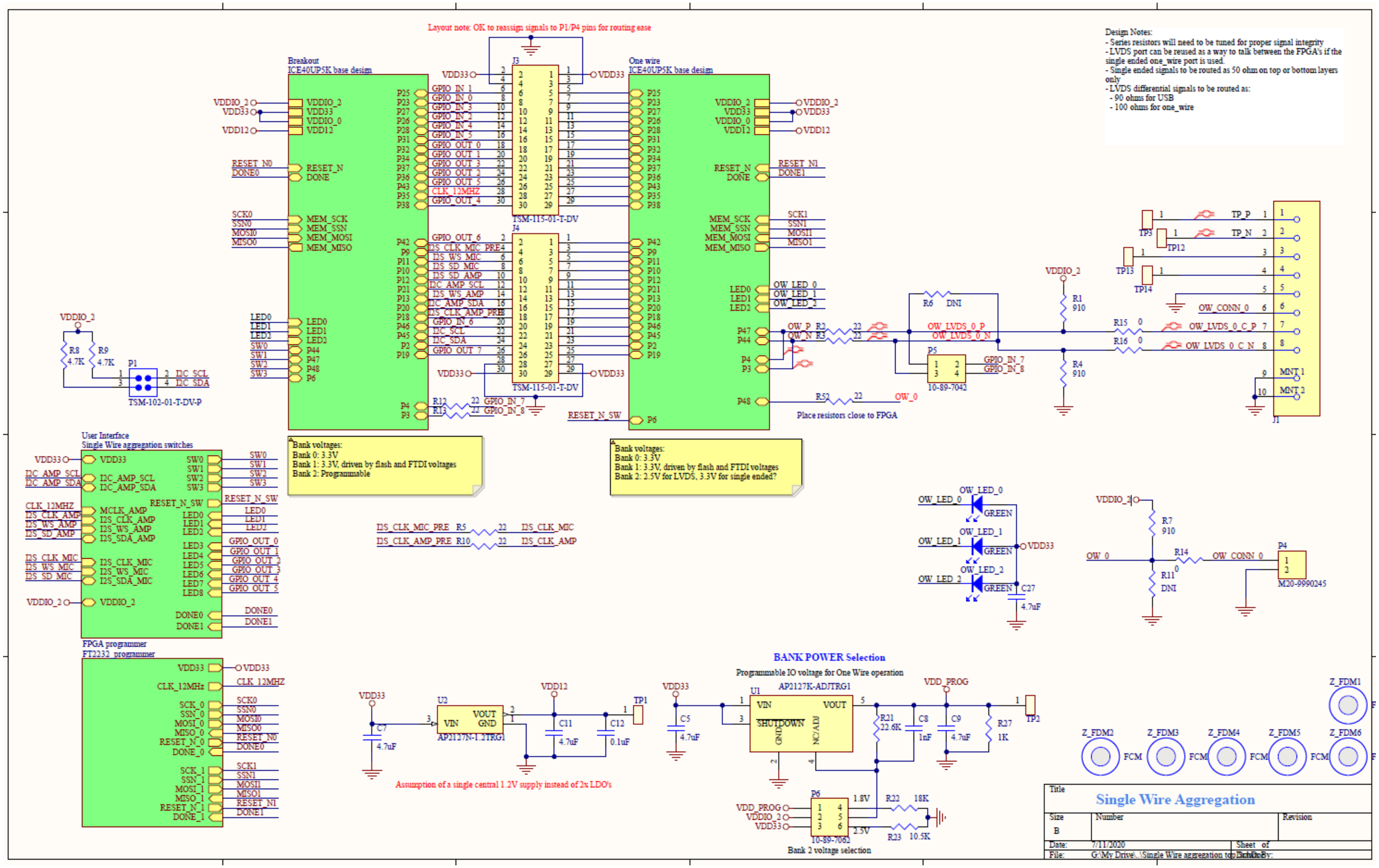
  For I2C2:
  - At Slave SWA Board, Press SW0. This causes the Slave Overhead FPGA to generate 9 I^{2}C commands to set-up, enable DAC on the Master SWA Board.
  - 1 kHz single tone or audio coming from the I2S microphone of the Master SWA board is observed on the audio receiver connected on the Master SWA board’s audio jack.
  - Pressing Slave SWA Board - SW0 again generates I^{2}C command to mute and unmute the Master SWA Board - DAC.

- I2S Signal Aggregation:

  For I2S1:

  - Master SWA FPGA Aggregate I2S Channel 1 to Slave SWA FPGA. It acts as a I2S slave. I2S data, I2S clock and I2S WS are being generated by Master Overhead FPGA that is send to Master SWA FPGA for aggregation. I2S sampling rate is at ~48 kHz.
  - Master Overhead FPGA generate I2S data. I2S data being sent to Master SWA FPGA are either a 1 kHz single tone or I2S data coming from Master SWA Board - I2S microphone. Pressing or switching SW2 at Master SWA board selects the I2S data being sent to Master SWA FPGA.
  - De-aggregated I2S data from Slave SWA board are sent to Slave Overhead FPGA to verify if its receiving a 1 kHz single tone. Same signal are sent to DAC so that you can verify the received audio through audio jack. LED0 on Slave SWA FPGA indicates the status of I2S data verification. Blinking LED0 means its receiving expected single tone data.
  - Switching SW2 at Slave SWA board resets I2S verification being done by Slave Overhead FPGA.
  - I2S Signals are also feedback to Master Overhead FPGA. After resetting Master Overhead FPGA, Master SWA LED0 is blinking which indicate it is sending 1 kHz single tone signal to SWA FPGA.
  - Switching Slave SWA Board – SW2 resets I2S verification.

Appendix A. Board Schematics



Layout note: OK to reassign signals to P1/P4 pins for routing ease

Design Notes:
- Series resistors will need to be tuned for proper signal integrity
- LVDS port can be reused as a way to talk between the FPGA's if the single ended one_wire port is used.
- Single ended signals to be routed as 50 ohm on top or bottom layers only
- LVDS differential signals to be routed as:
  - 90 ohms for USB
  - 100 ohms for one_wire

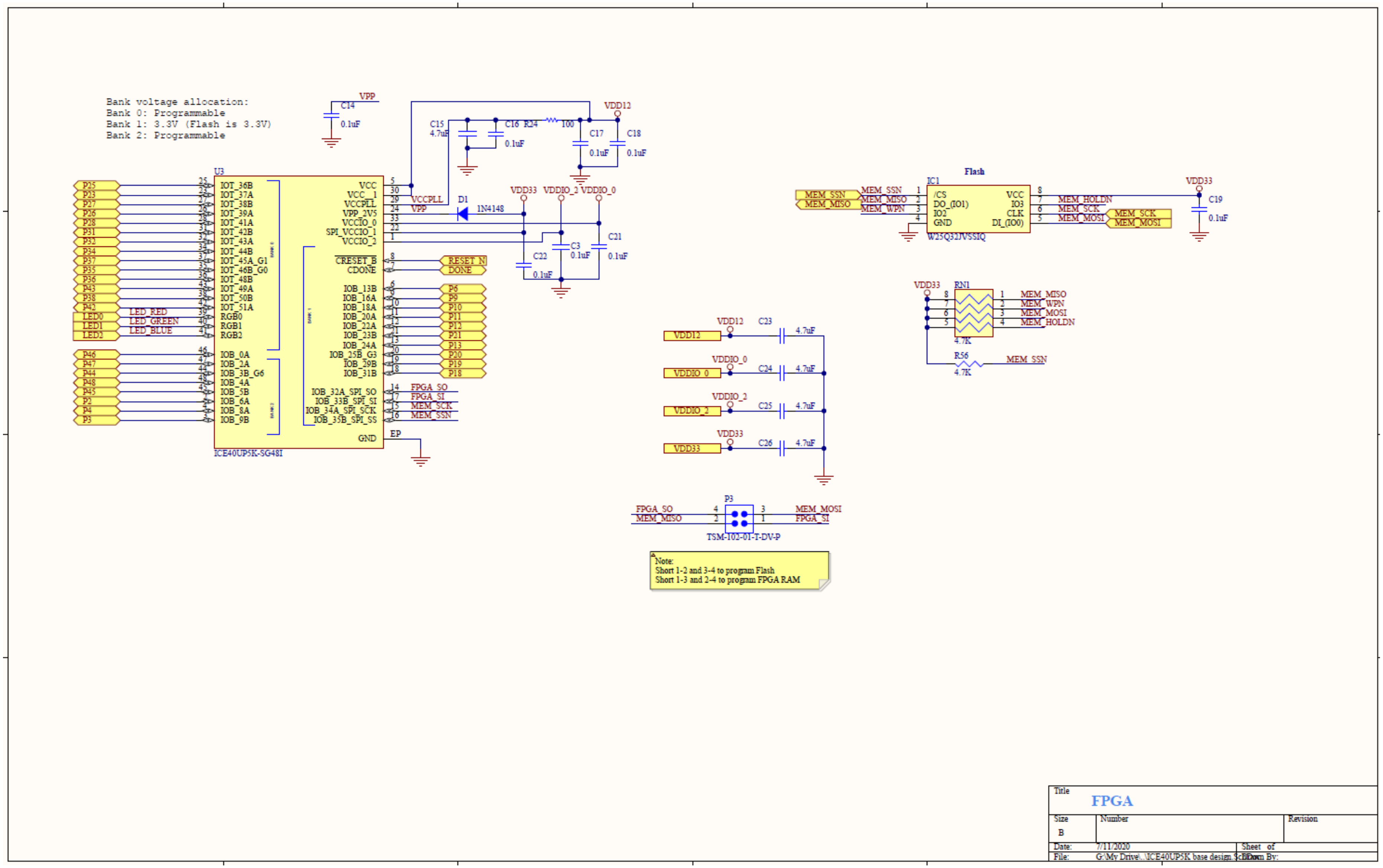
Bank voltages:
Bank 0: 3.3V
Bank 1: 3.3V, driven by flash and FTDI voltages
Bank 2: Programmable

Bank voltages:
Bank 0: 3.3V
Bank 1: 3.3V, driven by flash and FTDI voltages
Bank 2: 2.5V for LVDS, 3.3V for single ended?

Place resistors close to FPGA

Assumption of a single central 1.2V supply instead of 2x LDO's

Figure A.1. Single-Wire Aggregation Evaluation Board Schematics (Part 1). Note: OW (One Wire) is same with Single-Wire Link



Note:
Short 1-2 and 3-4 to program Flash
Short 1-3 and 2-4 to program FPGA RAM

Figure A.2. Single-Wire Aggregation Evaluation Board Schematics (Part 2)

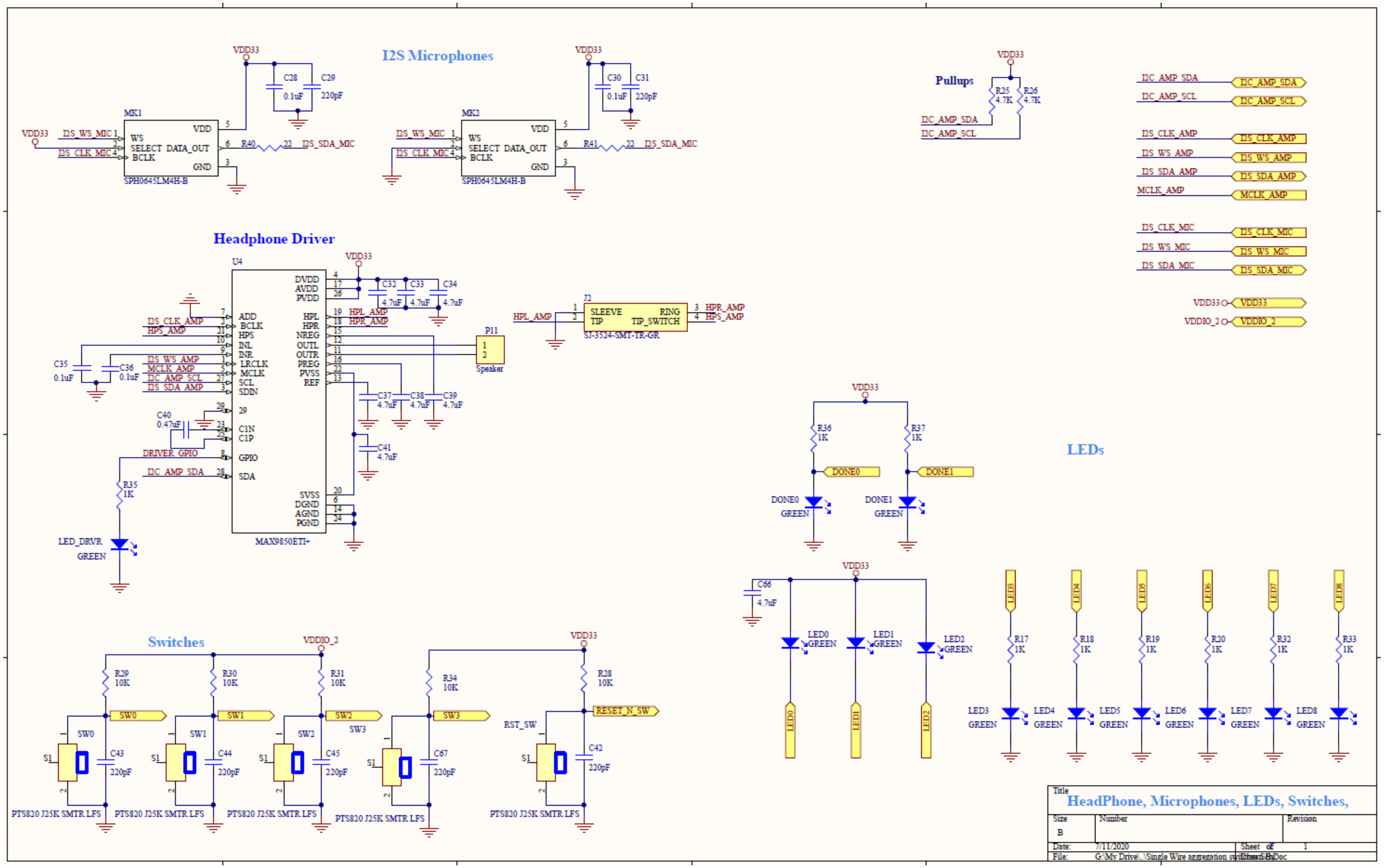


Figure A.3. Single-Wire Aggregation Evaluation Board Schematics (Part 3)

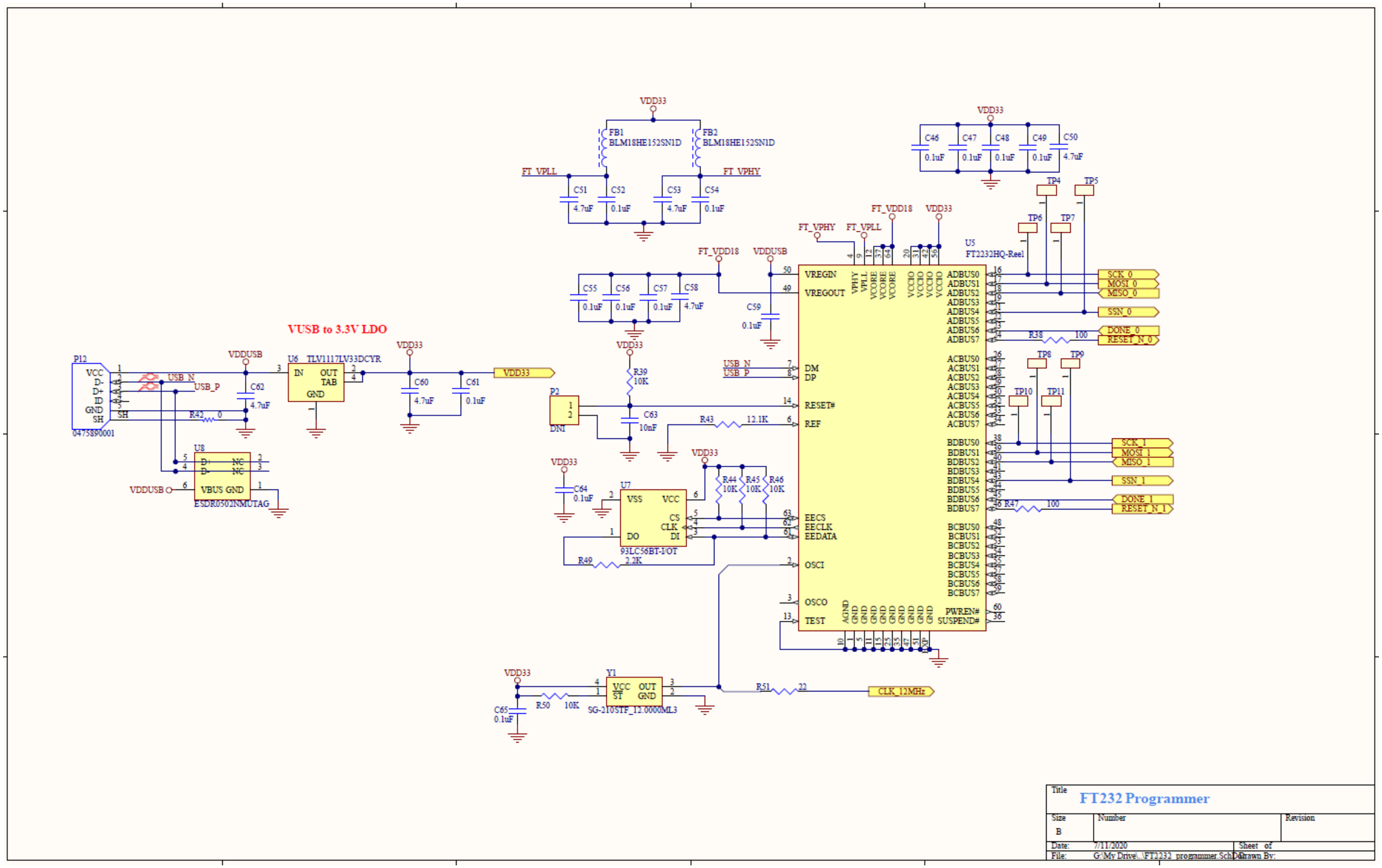


Figure A.4. Single-Wire Aggregation Evaluation Board Schematics (Part 4).

Supplemental Information

Single-Wire Aggregation device is based on the iCE40 UltraPlus FPGA. A variety of technical documents for the iCE40 UltraPlus family are available on the Lattice web site.

- iCE40 UltraPlus Family Data Sheet (FPGA-DS-02008)
- Lattice Single-wire Aggregation Design (FPGA-RD-02039)
- iCE40 Programming and Configuration (FPGA-TN-02001)
- iCE40 Hardware Checklist (TN1252)
- iCE40 LED Driver Usage Guide (TN1288)
- Thermal Management
- Package Diagrams
- Lattice design tools

Technical Support Assistance

For assistance, submit a technical support case at www.latticesemi.com/techsupport.

Revision History

Revision 1.0, September 2020

Section	Change Summary
All	Initial release.

LATTICE
SEMICONDUCTOR®

www.latticesemi.com