SPI_Slave
XulA2-LX9 + StickIt!-MB
with
Raspberry Pi 2B (RPi2B) Master
four wire
MOSI, MISO, SCLK, and SS
04/11/16

Raspberry Pi 2B RPi2B provides connections to the StickIt!-MB which provides the signals needed to communicate with the XulA2-LX9 or XulA2-LX25.

An external led is connected to T4 PM2 Xula2 CH 15 led<0>. Following the loading of the FPGA the external led will blink at 1 per sec. This is currently functioning correctly.

Four signals provide the SPI interface between the RPi2B and XulA2.

Xula2 CH 25 pin 19 GPIO10(SPI_MOSI)
Xula2 CH 24 pin 21 GPIO09 (SPI_MISO)
Xula2 CH 24 pin 23 GPIO11 (SPI_CLK)
Xula2 CH 08 pin 24 GPIO08_CE0

In the build module of the xula2_blinky_spi_slave is where the pins are mapped to RPi2B connector P1.

```
brd.add_port_name('led', 'pm2', slice(0, 8))
#xula2 chan25 BCM10_MOSI   ----> RPI_GPIO_P1_19 /* MOSI */
brd.add_port('mosi', 'F2')
#xula2 chan24 BCM09_MISO  ---->  RPI_GPIO_P1_21 /* MISO */
brd.add_port('miso', 'F1')
#xula2 chan23 BCM11_SCLK  ----> RPI_ GPIO_P1_23  /* CLK */
brd.add_port('sck', 'H2')
#xula2 chan08 BCM08_CE0   ----> RPI_GPIO_P1_24 /* CE0 */
brd.add_port('ss', 'J14')
```

The above information is what rhea uses to create the xula2.ucf

The rhea package creates several files xula2.v xula2.ucf below.

```
#
NET "miso" LOC = "F1" ;
NET "led<0>" LOC = "T4" ;
NET "led<1>" LOC = "R1" ;
NET "led<2>" LOC = "M1" ;
NET "led<3>" LOC = "J4" ;
NET "led<4>" LOC = "R2" ;
NET "led<5>" LOC = "M2" ;
NET "led<6>" LOC = "K3" ;
NET "led<7>" LOC = "H1" ;
```

```
NET "ss" LOC = "J14" ;
NET "clock" LOC = "A9" ;
NET "mosi" LOC = "F2" ;
NET "sck" LOC = "H2" ;
#
NET "clock" TNM_NET = "clock";
TIMESPEC "TS_clock" = PERIOD "clock" 83.3333333 ns HIGH 50%;
#
```

In RPi2B bcm2835 library. This is where the pins for the spi are assigned

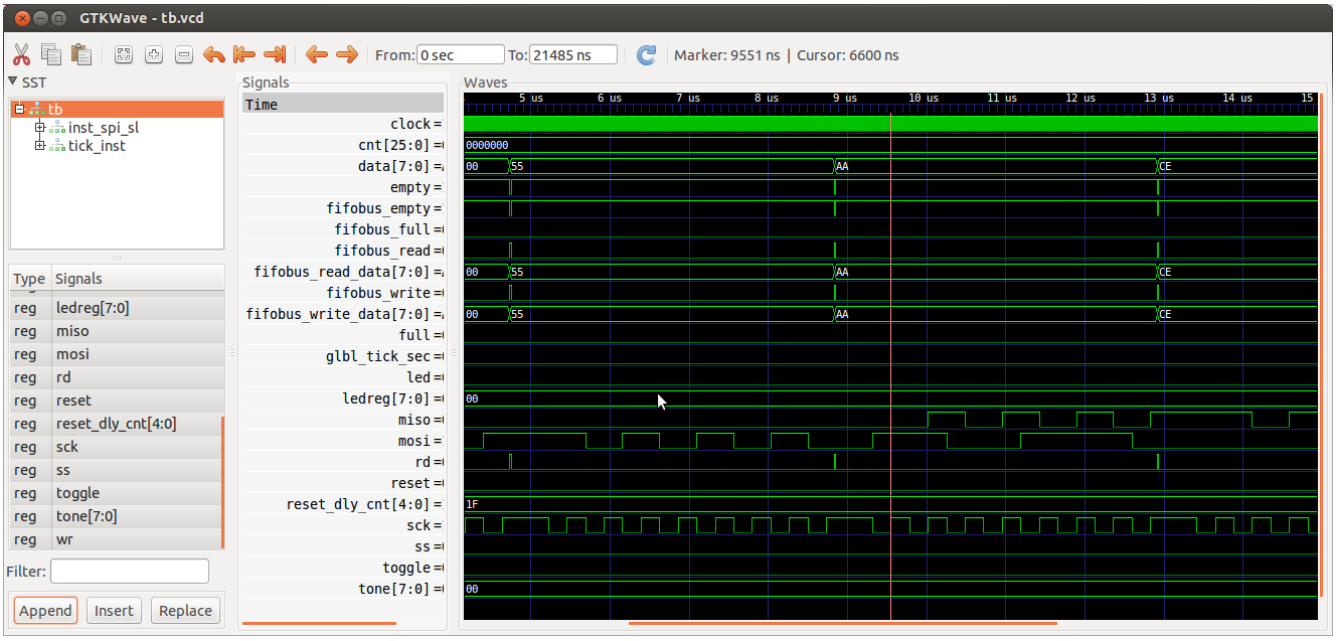
```
void bcm2835_spi_begin(void)
{
    volatile uint32_t* paddr;

    /* Set the SPI0 pins to the Alt 0 function to enable SPI0 access on them */
    bcm2835_gpio_fsel(RPI_GPIO_P1_26, BCM2835_GPIO_FSEL_ALT0); /* CE1 */
    bcm2835_gpio_fsel(RPI_GPIO_P1_24, BCM2835_GPIO_FSEL_ALT0); /* CE0 */
    bcm2835_gpio_fsel(RPI_GPIO_P1_21, BCM2835_GPIO_FSEL_ALT0); /* MISO */
    bcm2835_gpio_fsel(RPI_GPIO_P1_19, BCM2835_GPIO_FSEL_ALT0); /* MOSI */
    bcm2835_gpio_fsel(RPI_GPIO_P1_23, BCM2835_GPIO_FSEL_ALT0); /* CLK */

    /* Set the SPI CS register to the some sensible defaults */
    paddr = bcm2835_spi0 + BCM2835_SPI0_CS/4;
    bcm2835_peri_write(paddr, 0); /* All 0s */

    /* Clear TX and RX fifos */
    bcm2835_peri_write_nb(paddr, BCM2835_SPI0_CS_CLEAR);
}
```

In file spi.c after calling the bcm2835_spi_begin(); above the bit order is set to MSBFIRST

```
    bcm2835_spi_setBitOrder(BCM2835_SPI_BIT_ORDER_MSBFIRST);      // The default
    bcm2835_spi_setDataMode(BCM2835_SPI_MODE0);                   // The default
    bcm2835_spi_setClockDivider(BCM2835_SPI_CLOCK_DIVIDER_65536); // The default
    bcm2835_spi_chipSelect(BCM2835_SPI_CS0);                      // The default
    bcm2835_spi_setChipSelectPolarity(BCM2835_SPI_CS0, LOW);      // the default
```

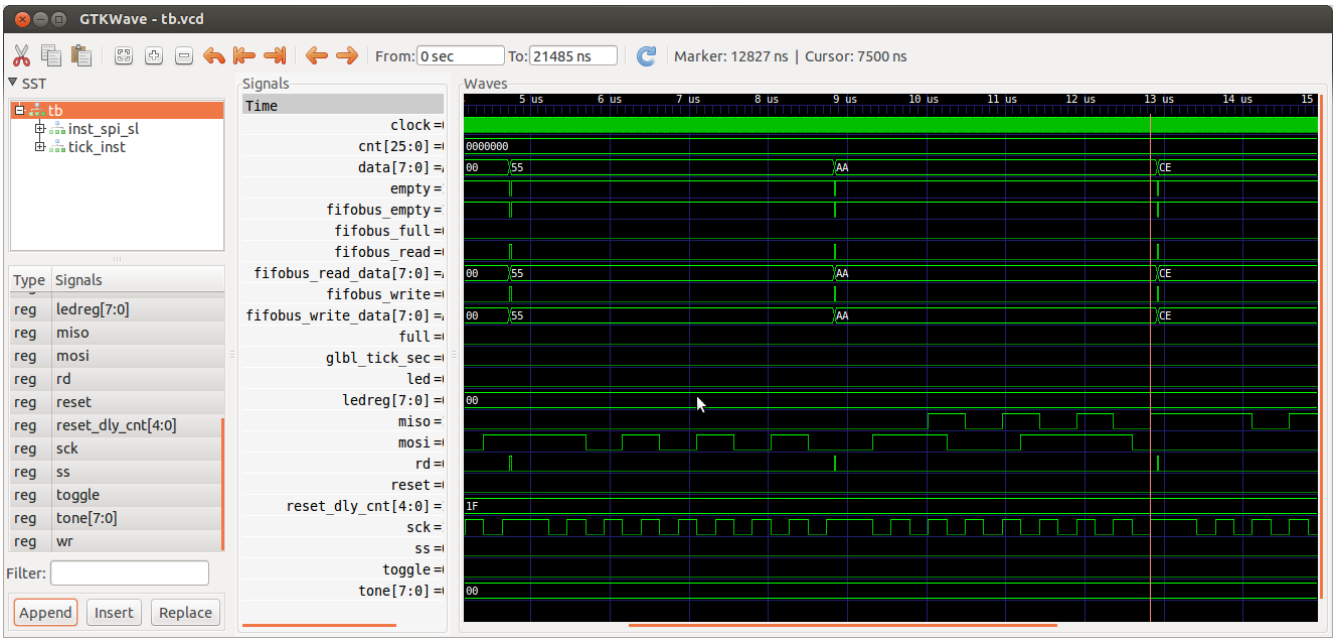
To run the simulation ***"python xula2_blinky_spi_slave.py –trace"*** which traces the signals of def tb(led, clock, mosi, miso, sck, ss, reset=None): and creates the ***"tb.vcd"*** file that can be viewed with ***"gtkwave tb.vcd"***

The RPi2B is set to send MSBFIRST. In the simulation below the signal mosi MSB of the value 0xCE 11001110 is at 9551 ns on the raising edge of the signal sck.



In the simulation below the signal mosi LSB of 0xCE 11001110 is

at 12827 ns the raising edge of the signal sck.



This appears to match the RPi2B sending MSBFIRST.

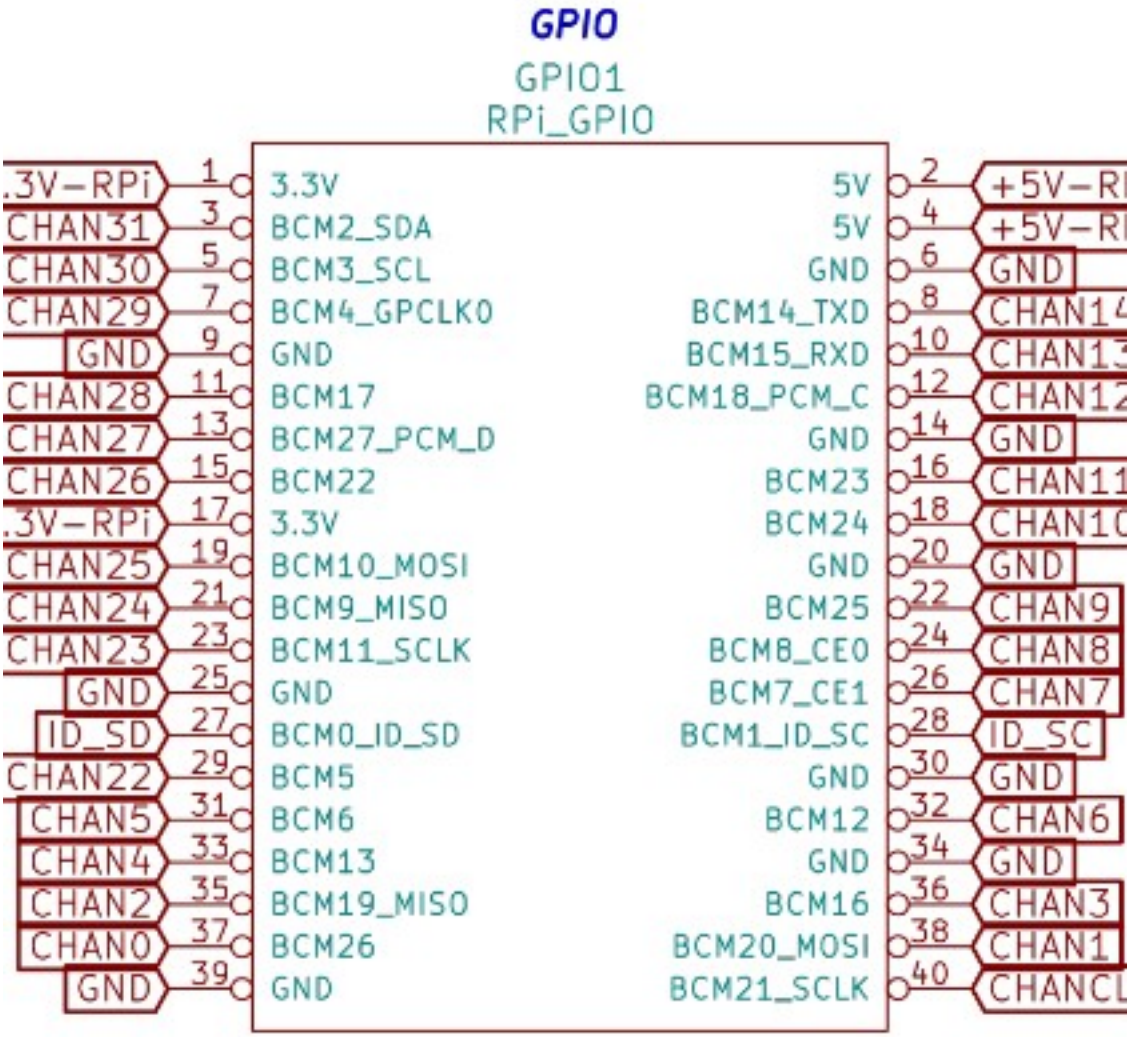
In addition rhea creates the xula2.tcl file.

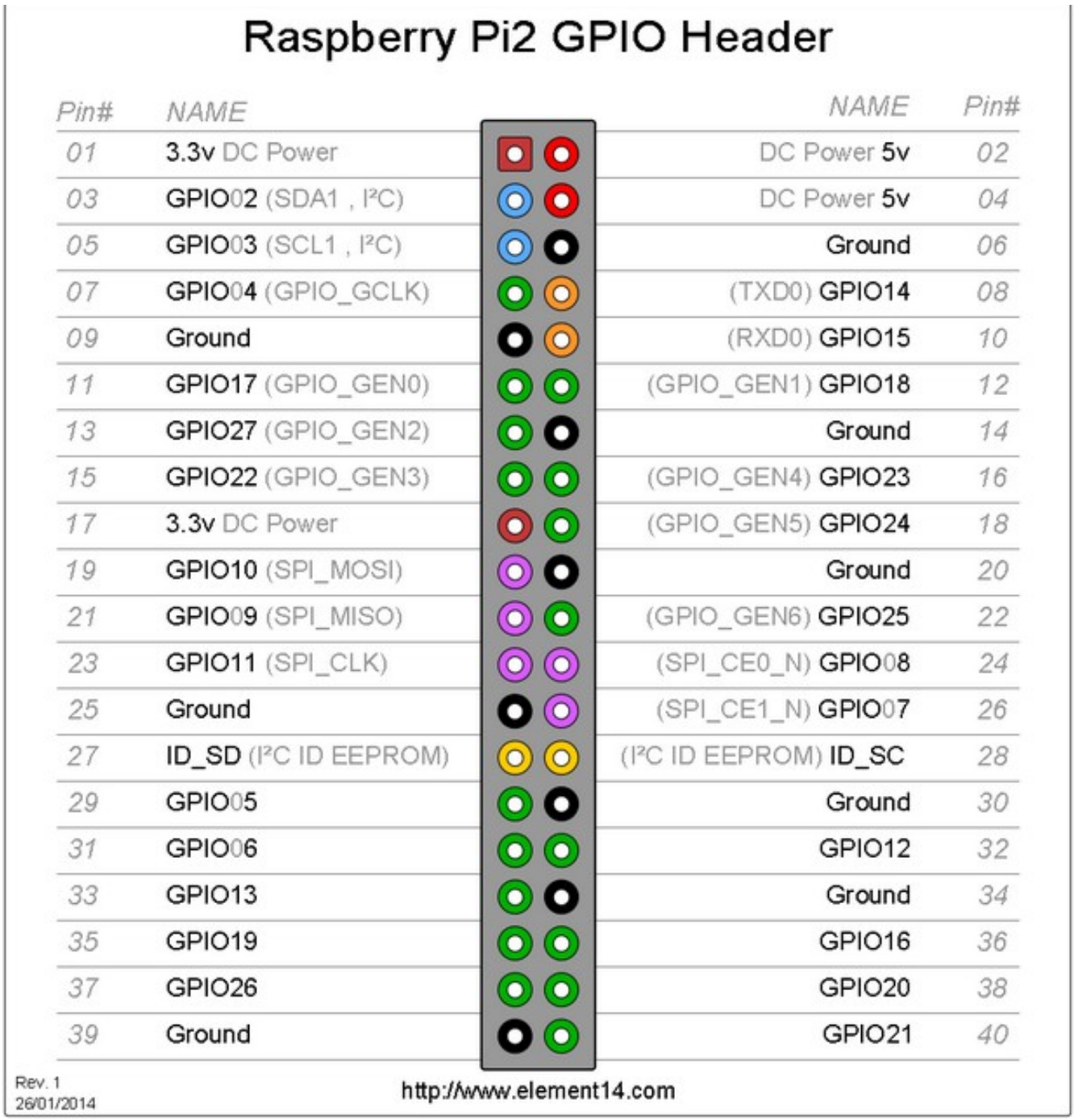
```
#
#
# ISE implementation script
# create: Sun, 10 Apr 2016 12:13:22 +0000
# by: xula2_blinky_spi_slave.py
#
#
# set compile directory:
set compile_directory .
set top_name xula2
set top xula2
# set Project:
set proj xula2
# change to the directory:
cd xilinx/
# set ucf file:
set constraints_file xula2.ucf
# set variables:
project new xula2.xise
project set family spartan6
project set device XC6SLX9
project set package FTG256
project set speed -2

# add hdl files:
xfile add xula2.ucf
xfile add xula2.v
# test if set_source_directory is set:
if { ! [catch {set source_directory $source_directory}]} {
  project set "Macro Search Path"
 $source_directory -process Translate
}
project set "Create Binary Configuration File" "true" -process "Generate Programming File"
 project set "FPGA Start-Up Clock" "JTAG Clock" -process "Generate Programming File"
# run the implementation:
process run "Synthesize"
```

```
process run "Translate"
process run "Map"
process run "Place & Route"
process run "Generate Programming File"
# close the project:
project close
```

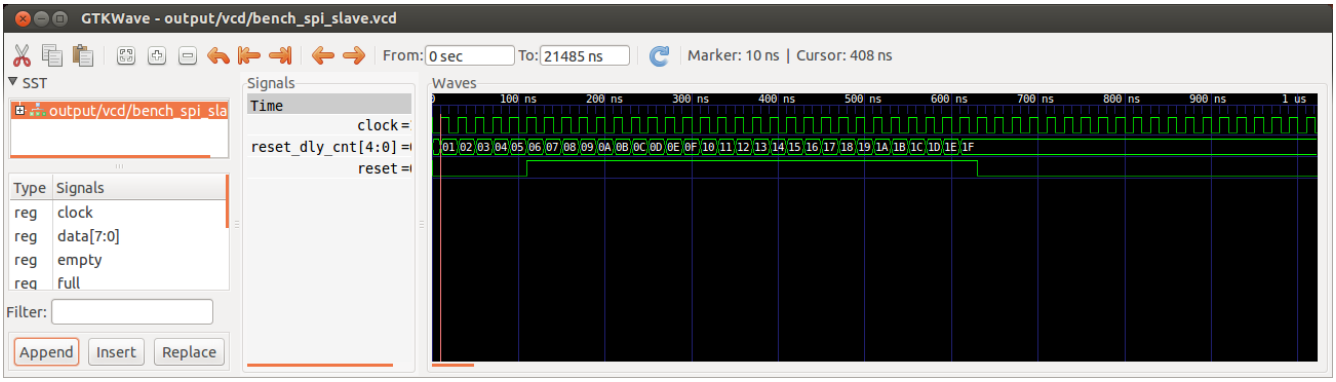
Here are the FPGA pins connected to each channel for both the XuLA and XuLA2 Boards:

XuLA	XuLA2	J1	Pin#	J1		XuLA2	XuLA
AN1	AN1		1	21	RESET	RESET	RESET
AN0	AN0		2	22		+1.2V	+1.2V
+5V	+5V	+5V	3	23	GND	GND	GND
P88	A2	CH31	4	24		+3.3V	+3.3V
P89	B2	CH30	5	25	CH14	B15	P84
P93	B1	CH29	6	26	CH13	B16	P83
P94	C1	CH28	7	27	CH12	C15	P82
P97	E2	CH27	8	28	CH11	C16	P73
P3	E1	CH26	9	29	CH10	F16	P72
P4	F2	CH25	10	30	CH9	F15	P68
P7	F1	CH24	11	31	CH8	J14	P62
P12	H2	CH23	12	32	CH7	J16	P61
P13	H1	CH22	13	33	CH6	K16	P57
P19	J4	CH21	14	34	CH5	K15	P56
P20	K3	CH20	15	35	CH4	M16	P52
P21	M1	CH19	16	36	CH3	M15	P50
P32	M2	CH18	17	37	CH2	R16	P39
P33	R1	CH17	18	38	CH1	R15	P37
P34	R2	CH16	19	39	CH0	R7	P36
P35	T4	CH15	20	40	CHCLK	T7	P44


GPIO
GPIO1
RPi_GPIO
.3V-RPi
CHAN31
CHAN30
CHAN29
GND
CHAN28
CHAN27
CHAN26
.3V-RPi
CHAN25
CHAN24
CHAN23
GND
ID_SD
CHAN22
CHAN5
CHAN4
CHAN2
CHAN0
GND
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
3.3V
BCM2_SDA
BCM3_SCL
BCM4_GPCLK0
GND
BCM17
BCM27_PCM_D
BCM22
3.3V
BCM10_MOSI
BCM9_MISO
BCM11_SCLK
GND
BCM0_ID_SD
BCM5
BCM6
BCM13
BCM19_MISO
BCM26
GND
5V
5V
GND
BCM14_TXD
BCM15_RXD
BCM18_PCM_C
GND
BCM23
BCM24
GND
BCM25
BCM8_CE0
BCM7_CE1
BCM1_ID_SC
GND
BCM12
GND
BCM16
BCM20_MOSI
BCM21_SCLK
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
+5V-R
+5V-R
GND
CHAN14
CHAN13
CHAN12
GND
CHAN11
CHAN10
GND
CHAN9
CHAN8
CHAN7
ID_SC
GND
CHAN6
GND
CHAN3
CHAN1
CHANCL



Creating a software reset for the XulA2-LX with StickIt!-MB.

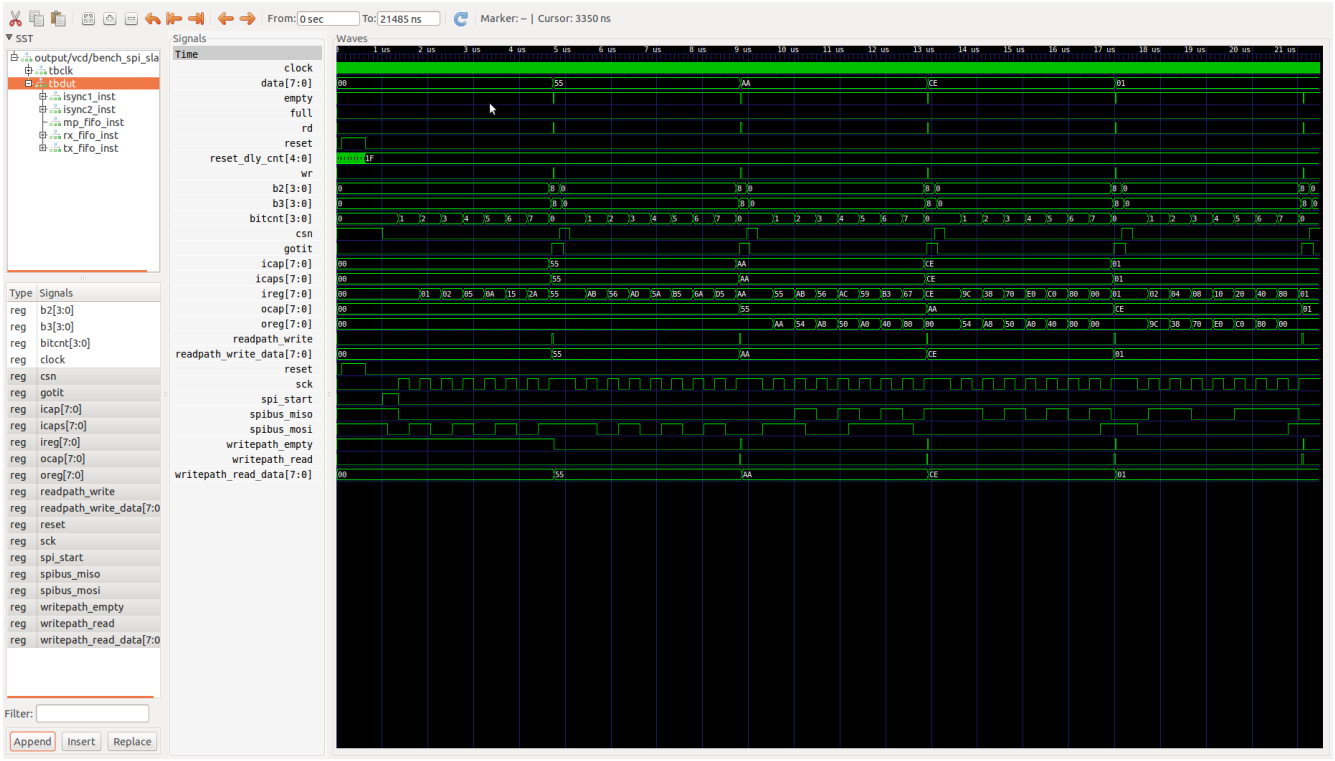


SPI_Slave software is provided with rhea package.

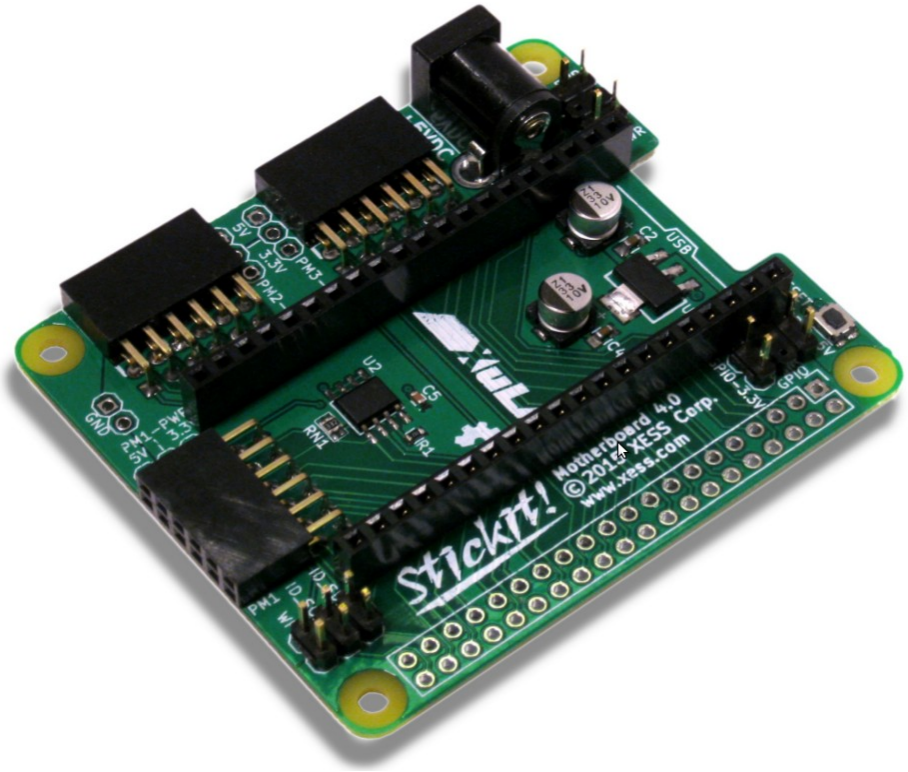
from rhea.system import Global, Clock, Reset, FIFOBus, Signals
- /home/vidal/wkg/rhea/rhea/system/glbl.py
  - Global
- /home/vidal/wkg/rhea/rhea/system/clock.py
  - Clock
- /home/vidal/wkg/rhea/rhea/system/reset.py
  - Reset
- /home/vidal/wkg/rhea/rhea/system/stream/fifobus.py
  - FIFOBus
- /home/vidal/wkg/rhea/rhea/system/hwtypes.py
  - Signals

from rhea.cores.spi import SPIBus, spi_slave_fifo
- /home/vidal/wkg/rhea/rhea/cores/spi/spi.py
  - SPIBus
- /home/vidal/wkg/rhea/rhea/cores/spi/spi_fifo_slave.py
  - spi_slave_fifo

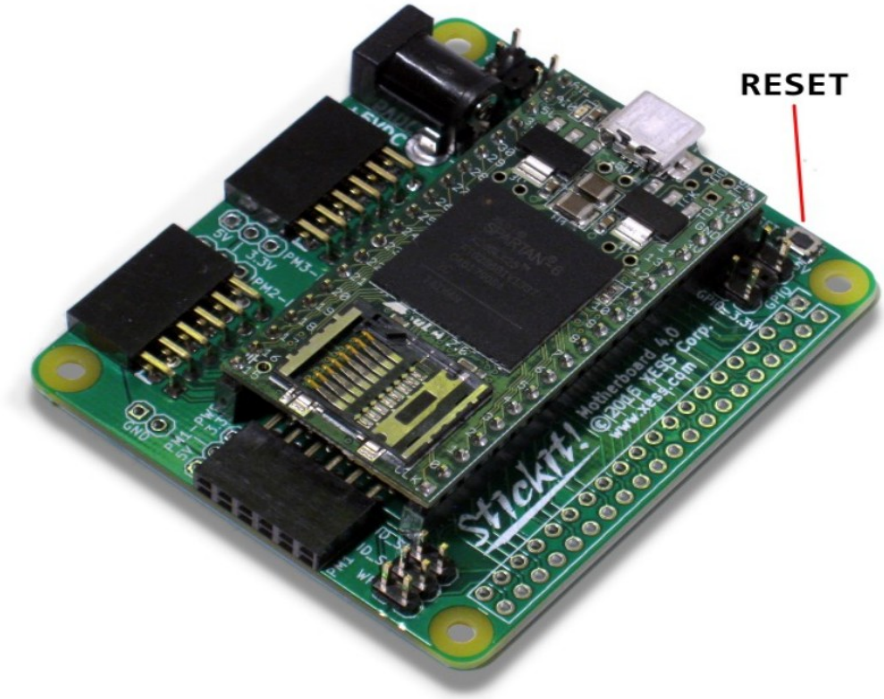
from rhea.cores.misc import glbl_timer_ticks
- /home/vidal/wkg/rhea/rhea/cores/misc/glbl_timer_ticks.py
  - glbl_timer_ticks


From: 0 sec
To: 21485 ns
Marker: – | Cursor: 3350 ns
SST
output/vcd/bench_spi_sla
tbclk
tbdut
isync1_inst
isync2_inst
mp_fifo_inst
rx_fifo_inst
tx_fifo_inst
Type Signals
reg b2[3:0]
reg b3[3:0]
reg bitcnt[3:0]
reg clock
reg csn
reg gotit
reg icap[7:0]
reg icaps[7:0]
reg ireg[7:0]
reg ocap[7:0]
reg oreg[7:0]
reg readpath_write
reg readpath_write_data[7:0
reg reset
reg sck
reg spi_start
reg spibus_miso
reg spibus_mosi
reg writepath_empty
reg writepath_read
reg writepath_read_data[7:0
Filter:
Append
Insert
Replace
Signals
Time
clock
data[7:0]
empty
full
rd
reset
reset_dly_cnt[4:0]
wr
b2[3:0]
b3[3:0]
bitcnt[3:0]
csn
gotit
icap[7:0]
icaps[7:0]
ireg[7:0]
ocap[7:0]
oreg[7:0]
readpath_write
readpath_write_data[7:0]
reset
sck
spi_start
spibus_miso
spibus_mosi
writepath_empty
writepath_read
writepath_read_data[7:0]
Waves

The StickIt!-MB provides a 3 PMOD connectors a 40 pin connector to the RPi2B. In addition the a socket to support either a XulA2-LX9 or XulA2-LX25 FPGA.



XulA2 mounted on StickIt!-MB.



Create a bit file to make the xula2 FPGA provide a jumper on

pins 19 GPIO10(SPI_MOSI) & pin 21 GPIO09 (SPI_MISO).

Set the PATH to ISE WEB PACK

. /opt/Xilinx/14.6/ISE_DS/settings64.sh

vidal@ws009:~/wkg/jpeg-2000-test/xula2_fpga/jumper_tests/spi$
python jumper_xula2_spi.py –build

vidal@ws009:~/wkg/jpeg-2000-test/xula2_fpga/jumper_tests/spi$ cp xilinx/xula2.bit
xilinx/jumper_spi.bit

vidal@ws009:~/wkg/jpeg-2000-test/xula2_fpga/jumper_tests/spi$
python jumper_xula2_spi.py –build

pi@raspberrypi2-146 ~/bcm2835-1.45/examples/spi $
xsload --usb 0 --fpga ~/jpeg-2000/xula2_fpga/
jumper_tests/spi/xilinx/jumper_spi.bit

pi@raspberrypi2-146 ~/bcm2835-1.45/examples/spi $
gcc -o spi spi.c -l bcm2835

pi@raspberrypi2-146 ~/bcm2835-1.45/examples/spi $ sudo ./spi
Sent to SPI: 0x23. Read back from SPI: 0x23.
pi@raspberrypi2-146 ~/bcm2835-1.45/examples/spi

Appendix C. XulA2-LX25 & XulA2-LX9 freatures.

LX25

XuLA2-LX25

24051	*Build bigger circuits!*
1165	*Use larger buffers!*
38	*Do even more DSP!*
4	*The same digital clock managers...*
2	*...but also phase-locked loops!*
16M x 16	*More external RAM!*
8	*More Flash for bigger bitstreams!*
Yes	*Store lots of data (or an OS)!*
33 / 33	*Every pin is now an I/O pin!*
51 x 25	*Same footprint!*
9	*Gained a bit of weight.*
Yes	*Still completely open!*
$119.00	

LX9

XuLA2-LX9

9152	*Build bigger circuits!*
666	*Use larger buffers!*
16*	*The LX9 has multiplier-accumulators!*
4	*The same digital clock managers...*
2	*...but also phase-locked loops!*
16M x 16	*More external RAM!*
8	*More Flash for bigger bitstreams!*
Yes	*Store lots of data (or an OS)!*
33 / 33	*Every pin is now an I/O pin!*
51 x 25	*Same footprint!*
9	*Gained a bit of weight.*
Yes	*Still completely open!*
$69.00	*Costs a bit more, does a lot more!*

LX25

- Open-source design
- XC6SLX25 FPGA
- 32 MByte SDRAM
- 8 Mbit Flash
- microSD card socket
- 3.3 & 1.2V regulators
- 40-pin interface
- 12 MHz oscillator
- PIC 18F14K50 micro
- USB 2.0 port
- Auxiliary JTAG port
- Works with the XSTOOLs software
- Works with XILINX ISE and WebPACK
- Works with XILINX iMPACT and ChipScope (requires Xilinx JTAG cable)

LX9

- Open-source design
- XC6SLX9 FPGA
- 32 MByte SDRAM
- 8 Mbit Flash
- microSD card socket
- 3.3 & 1.2V regulators
- 40-pin interface
- 12 MHz oscillator
- PIC 18F14K50 micro
- USB 2.0 port
- Auxiliary JTAG port
- Works with the XSTOOLs software
- Works with XILINX ISE and WebPACK
- ***Works with XILINX iMPACT and ChipScope (requires Xilinx JTAG cable)***