

***SPI\_Slave***  
***Xula2-LX9 + StickIt!-MB***  
***with***  
***Raspberry Pi 2B (RPi2B) Master***  
***four wire***  
***MOSI, MISO, SCLK, and SS***  
***04/18/16***

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

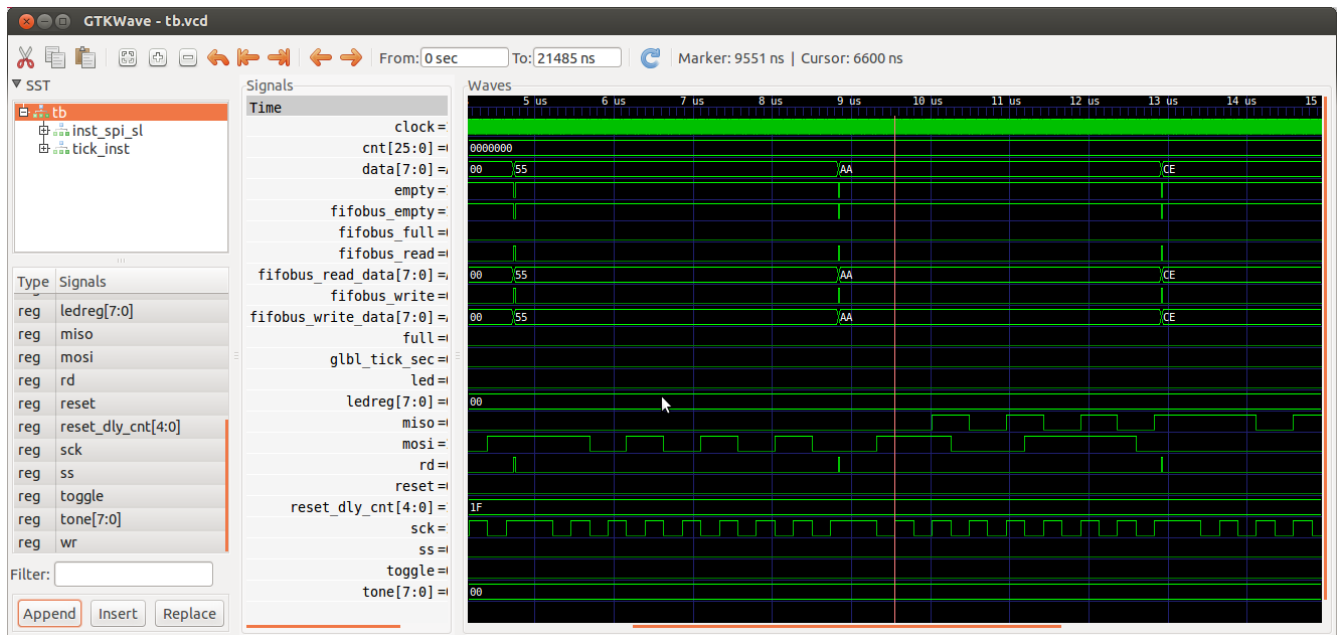
```

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

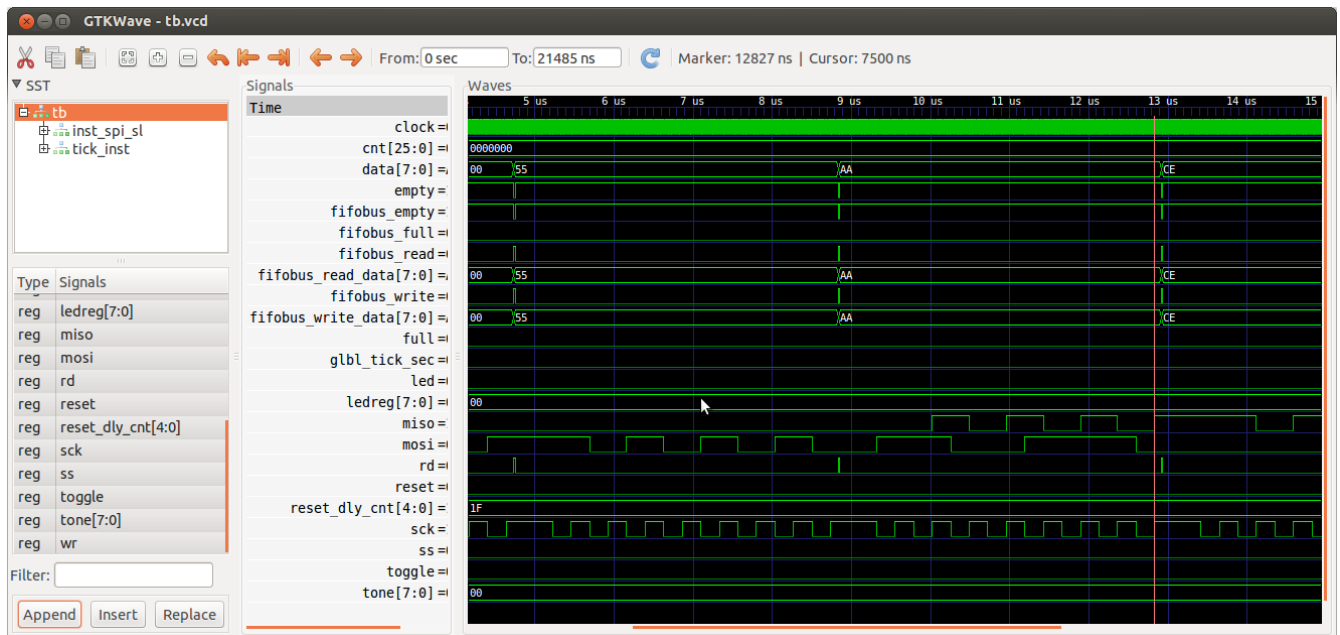
```

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

STICKIT-MB RPI2B						
XuLA2	CHAN	Rpi I/O	Rpi I/O		CHAN	XuLA2
		+3.3V	1 2	+5V		
A2	CH31	BCM2_SDA	3 4	+5V		
B2	CH30	BCM2_SCL	5 6	GND		
B1	CH29	BCM4_GPCLK0	7 8	BCM14_TXD	CH14	B15
GND		GND	9 10	BCM15_RXD	CH13	B16
C1	CH28	BCM17	11 12	BCM18_PCM_C	CH12	C15
E2	CH27	BCM27_PCM_D	13 14	GND		GND
E1	CH26	BCM22	15 16	BCM23	CH11	C16
		+3.3V	17 18	BCM24	CH10	F16
F2	CH25	BCM10_MOSI	19 20	GND		
F1	CH24	BCM9_MISO	21 22	BCM25	CH9	F15
H2	CH23	BCM11_SCLK	23 24	BCM8_CE0	CH8	J14
GND		GND	25 26	BCM7_CE1	CH7	J16
		BCM0_ID_SD	27 28	BCM1_ID_SC		
H1	CH22	BCM5	29 30	GND		GND
K15	CH5	BCM6	31 32	BCM12	CH6	K16
M16	CH4	BCM13	33 34	GND		GND
R16	CH2	BCM19_MISO	35 36	BCM16	CH3	M15
R7	CH0	BCM26	37 38	BCM20_MOSI	CH1	R15
GND		GND	39 40	BCM21_SCLK	CHANCLK	T7

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	1	3.3V	5V	2	+5V-RPi
CHAN31	3	BCM2_SDA	5V	4	+5V-RPi
CHAN30	5	BCM3_SCL	GND	6	GND
CHAN29	7	BCM4_GPCLK0	BCM14_TXD	8	CHAN14
GND	9	GND	BCM15_RXD	10	CHAN13
CHAN28	11	BCM17	BCM18_PCM_C	12	CHAN12
CHAN27	13	BCM27_PCM_D	GND	14	GND
CHAN26	15	BCM22	BCM23	16	CHAN11
.3V-RPi	17	3.3V	BCM24	18	CHAN10
CHAN25	19	BCM10_MOSI	GND	20	GND
CHAN24	21	BCM9_MISO	BCM25	22	CHAN9
CHAN23	23	BCM11_SCLK	BCM8_CEO	24	CHAN8
GND	25	GND	BCM7_CEO	26	CHAN7
ID_SD	27	BCM0_ID_SD	BCM1_ID_SC	28	ID_SC
CHAN22	29	BCM5	GND	30	GND
CHAN5	31	BCM6	BCM12	32	CHAN6
CHAN4	33	BCM13	GND	34	GND
CHAN2	35	BCM19_MISO	BCM16	36	CHAN3
CHAN0	37	BCM26	BCM20_MOSI	38	CHAN1
GND	39	GND	BCM21_SCLK	40	CHANCL

# Raspberry Pi2 GPIO Header

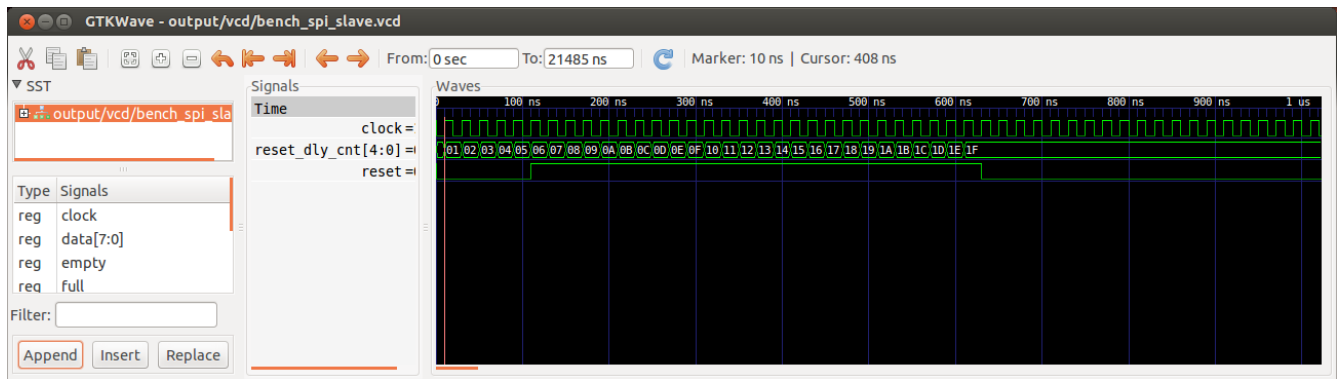
Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)		DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)		(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Rev. 1  
26/01/2014

<http://www.element14.com>

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
```

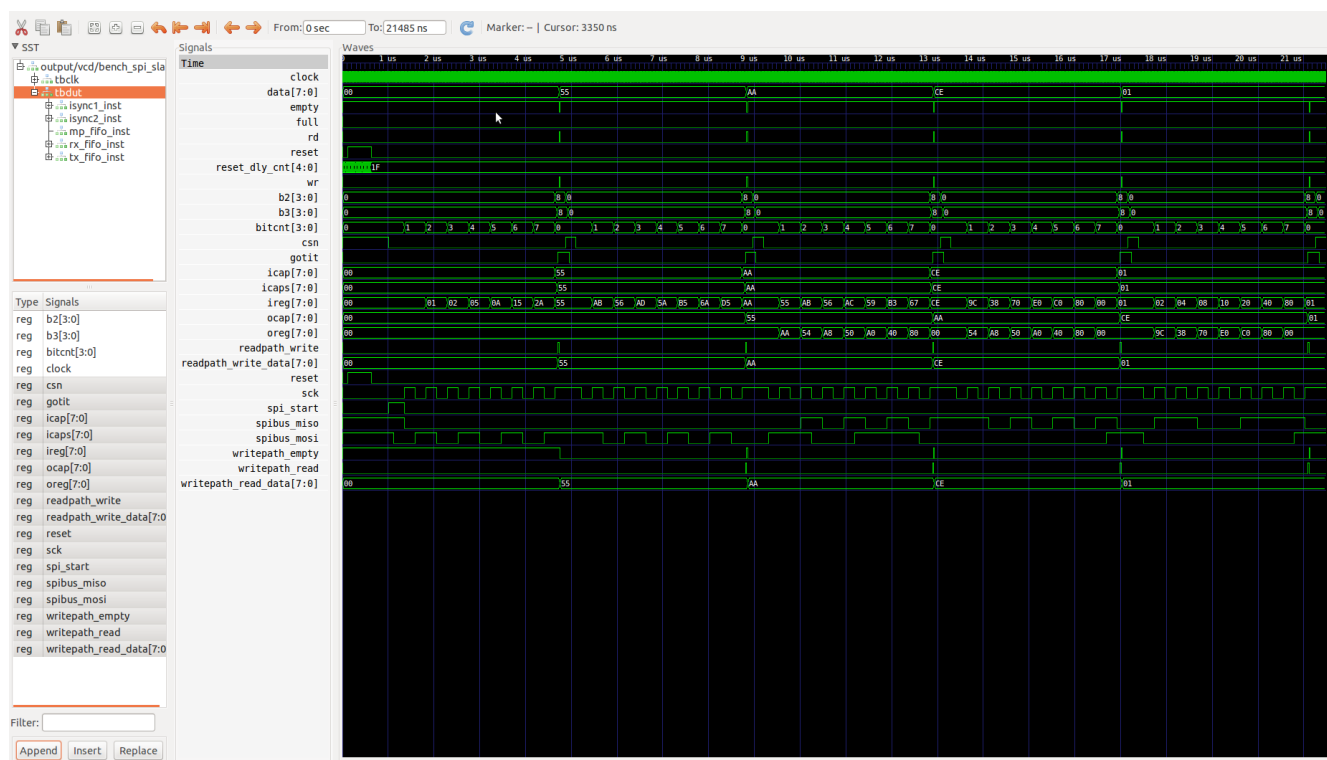
```
    /home/vidal/wkg/rhea/rhea/cores/spi/spi_fifo_slave.py
```

```
    spi_slave_fifo
```

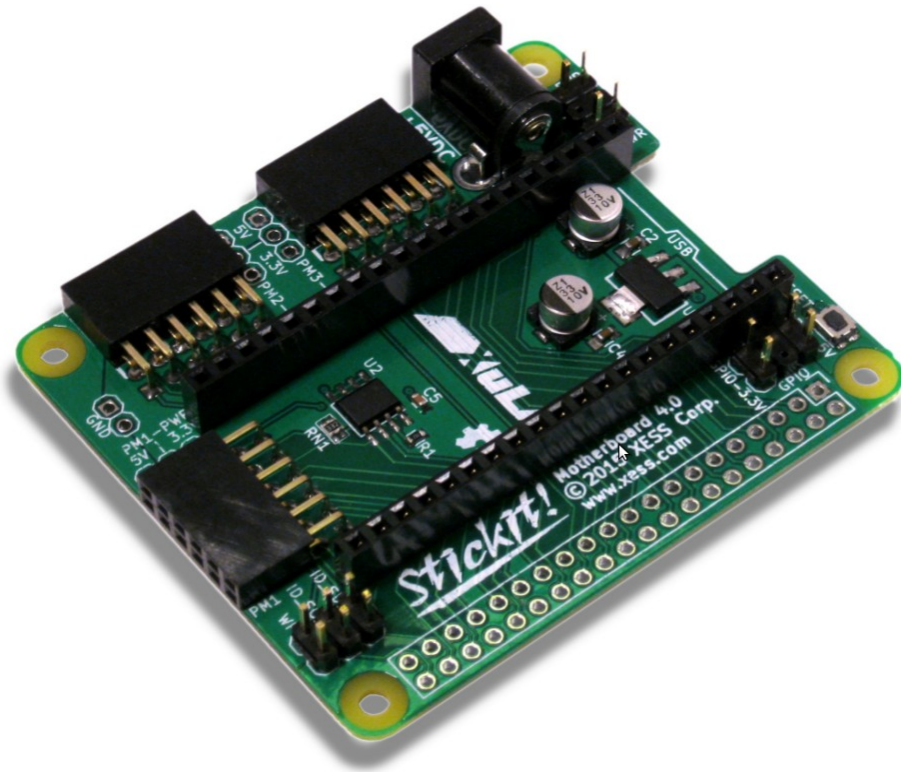
```
from rhea.cores.misc import glbl_timer_ticks
```

```
    /home/vidal/wkg/rhea/rhea/cores/misc/glbl_timer_ticks.py
```

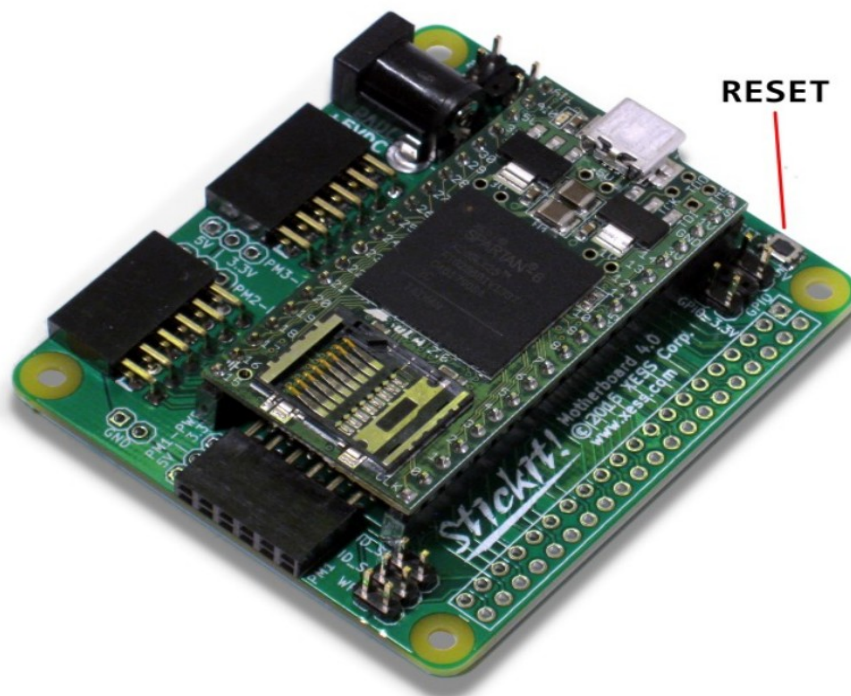
```
    glbl_timer_ticks
```



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 -I 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 features.*

## 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)**