CHAPTER 6

HARDWARE TEST

6.0 Introduction

The simulation covered in the previous chapter is not adequate to assure the designed circuit will function as intended. Therefore, there is a need to verify the design implementation on a real hardware. This chapter will guide the reader to the methods of testing the real FPGA that has been designed using Xilinx Foundation Series 2i (XSE2i). The hardware test will be using XSTOOLs and XS40 V1.1 prototyping board provided by XESS Corporation. The summary of devices used in the test are stated as below:

1. XS40 V1.1 prototyping board with XC4005E Xilinx FPGA chip and 12MHz resonator
2. XSTOOLS
   - GXSLOAD
   - GXSPORT
3. Hewlett Packard 54645D Mixed Signal Oscilloscope
4. K and H, GL-24 Project Boards
5. Parallel Downloading Cables
6. Jumper wires, resistors
7. PC: TOSHIBA Pentium III with Windows 98

8. VINSTAR NA-3R IC Regulated AC-DC Power Adapter (center positive, 9V DC output, 2.1mm female)

This chapter is organized in such way to present the description of the devices used in the test followed by the test setup of the hardware implementation. The hardware test continues with the discussion on each of the waveform captured using Hewlett Packard 54645D Mixed Signal Oscilloscope.

6.1 XS40 Prototyping Board Version 1.1 (XS40 V1.1)

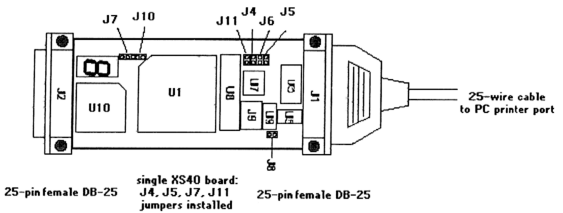


Figure 6.1: XS40 Board V1.1

Figure 6.1 shows XS40 V1.1 Prototyping Board with XC4005E, PC84 (U1) 9k-gates Xilinx FPGA chip [XEng, 1997]. This FPGA board is provided by XESS Corporation to download the implemented design (the UART) that has been designed previously using XSE2i. To have a clearer architecture of the prototyping board, the reader is advised to refer to the schematic architecture of the XS40 board, which located

at **Appendix 6**. The voltage regulator of XS40 will regulate the supplied voltage (i.e. 9V-DC, center positive) at jack J9 to 3.3V or 5V depending on the FPGA used. Since XC4000E type of FPGA will be used in this particular test case, the shunt (J8) should be removed to regulate the supplied voltage to 5V. Alternatively, power can also be applied directly through several pins of the XS Board. For the alternative way, a +5V VCC should be applied to pin 2 and grounded with the GND pin (pin 52).

6.2 Xilinx XC4005E FPGA Chip

Xilinx's XC4005E is manufactured underlying SRAM based process technology. SRAM cells are implemented as function generators to simulate combinatorial logic, control multiplexors and control routing resources. They are often used in reconfiguring computing applications where the functions of devices are dynamically changed. The capability of reconfiguration gives the system designer a new degree of freedom not available with any other types of logic. The capability to be programmed repeatedly without limits also suites the UART design.

Table 6.1: Summary of Xilinx XC4005E FPGA Chip

Device	Logic Cells	Max Logic Gates (No RAM)	Max. RAM Bits (No Logic)	Typical Gate Range (Logic and RAM)*	CLB Matrix	Total CLBs	Number of Flip-Flops	Max. User I/O
XC4005E /XL	466	5,000	6,272	3,000 - 9,000	14 x 14	196	616	112

The specification of Xilinx XC4005E FPGA Chip [Xilinx, 1999] is summarized in **Table 6.1**. As can be seen, the maximum logic gates without RAM the Xilinx XC4005E FPGA chip can occupy is 5,000 gates. The maximum logic gates are enough for the implementation of the UART with embedded BIST design since the intended design only needs low gates count (about 3000 gates).

6.3 XSTOOLs

To manipulate the logic characteristics of a XC4005E FPGA chip, some sort of tools are needed. XSTOOLs are tools provided by XESS Corporation and will be used (with its XS40 prototyping board) to test the designed UART. These tools will download and test the downloaded bit-steam (**.bit) files in order to make the FPGA works as intended. XSTOOLs [Xess, 2000] consist of GXSLOAD, GXSPORT, GXSTEST and GXSSETCLK.

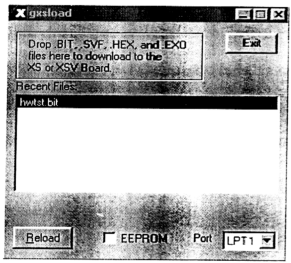


Figure 6.2: GXSLOAD

GXSLOAD [Xess, 2002] is a version of XSLOAD that uses a Windows graphical interface. XSLOAD is a command line program interface that will be used to configure the programmable device, download/upload the RAM, or set the oscillator frequency on either an XS40 Board (with XC4000 FPGA) or XS95 Board (with CPLD). To configure the programmable device, the user needs to create a bit stream file (with the extension **.bit). The bit stream file will be downloaded to the XS40 V1.1 by dropping the file into the GXSLOAD. GXSLOAD can also program an Atmel serial EEPROM if it is placed in the socket on the XS40 Board.

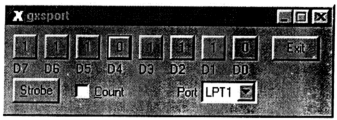


Figure 6.3: GXSPORT

GXSPORT is a much simpler program that just outputs a byte of binary bits onto the eight data pins of the parallel port. It will be used to force logic levels onto the input pins of the FPLD (FPGA or CPLD) to test a downloaded logic circuit.

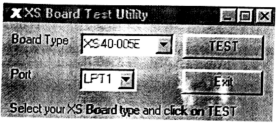


Figure 6.5: GXSTEST

GXSTEST is a test tool to test the XS95, XS40, and XSV (Virtex) Boards through a Windows graphical interface.

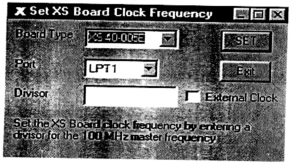


Figure 6.5: GXSSETCLK

GXSSETCLK is a tool to set the oscillator frequency of the XS95, XS40, or XSV Boards through a Windows graphical interface. However, XS40 V1.1 (which will be used in the test) does not included with a programmable oscillator, therefore it can only uses fixed 12MHz-clock resonator.

XS40 prototyping board and XSTOOLs are two main tools provided by Xess Corporation for FPGA test. In the coming section, the discussion will continue with the

initial test setup using these two tools and other devices. The initial test setup will provide the reader with the general information on how the test will be conducted.

6.4 Initial Test Setup

Generally, there are six steps involved in typical FPGA test using XS40 prototyping board. The steps can be summarized as follows:

1. Create a user constraint file (**.ucf)
2. Create an external test circuit (if needed)
3. Generate a bit stream file (**.bit) correspond to the user constraint file
4. Download the bit stream file to GXSLOAD
5. Drive input signals to input pins by using GXSPORT and external circuit
6. Observe output waveforms through Mixed Signal Oscilloscope

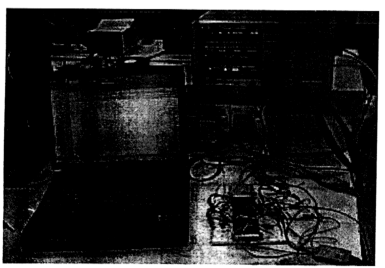


Figure 6.6: Test Setup

Figure 6.6 shows the initial hardware test setup for testing the FPGA. The computer note book will be used to download bit stream files (**.bit) to XC4005E FPGA by using the XSTOOLs and its parallel port. Since the XS40 does not draw power through the downloading cable of the PC parallel port, a 9V-DC center positive power adapter has to be switched on. The parallel-download cable then transmits the data needs by the XS40 to program the FPGA. (Caution: The adapter must has center positive polarity because inverting the polarity (i.e. center negative) will damage the FPGA since XS40 V1.1 did not has a diode to prevent such occurrence).

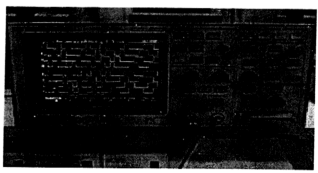


Figure 6.7: Hewlett Packard 54645D Mixed Signal Oscilloscope

In addition, Hewlett Packard 54645D Mixed-Signal-Oscilloscope (**Figure 6.7**) will be used to observe the output waveforms of the FPGA output's pins. The oscilloscope has up to 18-observation channel, where 16 digital waveforms and 2 analog waveforms can be observed simultaneously. Furthermore, the oscilloscope can automatically measures the frequency and the voltage outputs.

The previous sections have presented the main tools and devices for hardware test. It also covers the initial test setup by providing the reader with general test information on how the test will be conducted. Further discussion in the coming section will provide the reader with clearer view of the tools and devices used. The discussion will begin with the description of user constraint file (**.ucf) for specifying the FPGA pins. Then, the setup and test of a baud rate generator, LFSR, MISR and loop-back-control will be presented and discussed.

6.5 Baud Rate Generator Test

Figure 6.8: XS40 test setup

To start a hardware test, a bit stream (**.bit) file has to be created by XSE2i Project Manager implementation tools. However, it is not advisable to let the software to select the pin locations (if using XS40 board). This is because the software may locate the signals to pins that connected to conflicting resources on the XS40 board. For that reason User Constraint Files (**.ucf) will be used to select the appropriate pins.

6.5.1 User Constraint Files (**.ucf)

```
---------------------------------------------------------------------------------------------

NET n<3>     LOC = P57;
NET n<4>     LOC = P50;
NET n<5>     LOC = P58;
NET n<6>     LOC = P59;
NET n<7>     LOC = P62;
NET n<8>     LOC = P61;
NET n<9>     LOC = P60;
NET n<10>    LOC = P65;

NET n<11>   LOC = P49;
NET n<12>   LOC = P48;
NET n<13>   LOC = P47;
NET n<14>   LOC = P46;
NET n<15>   LOC = P45;

NET reseth LOC = P44;
NET hclk   LOC = P13;

NET bclkx16 LOC = P68;

---------------------------------------------------------------------------------------------
```

Above shows the user constraint file of Baud_r.ucf. This file will be used to specify appropriate pin location to produce a 16X clock (bclkx16). 'n' is a value that will program the baud rate generator's speed using a host clock (hclk). The "reseth" is a reset pin that initializes the baud rate generator. As can be observed, port n<3> to n<10> are located to free independent pins (pin 57, 50, 58, 59, 62, 61, 60, and 65). These pins need an external circuit (to drive input signals) at their input pins. On the other hand, port n<11> to n<15> and reseth are located to the pin that connected to the PC-parallel-port data outputs **(Appendix 6)**. The pin located at parallel port can use GXSPORT to drive the desired inputs to their pins (pin 44, 45, 46, 47, 48 and 49). A 12 MHz resonator

will be used to drive a clock signal to the host clock (hclk) at pin 13. The output of 16X clock for driving the internal transmitter logic (blkcX16) can be observed at pin 68 using mixed signal oscilloscope (Note: Please refer to **Appendix 2** for partial pad report of baud_r.pad).

6.5.2 Baud Rate Calculation

From the following equation, the value of n to be programmed into n<15, 14, 13,.....1, 0> are calculated.

$N = (16 \times 2^n \times BaudRate / Freq_{HCLK})$

= BRSEL1& BRSEL0 (in hexadecimal)

Solving for N when a maximum baud rate of 115,200 baud is to be used, accum_size generic = 16 (i.e. accumulator is 17 bits wide) and a host clock frequency of 12MHz:

$N = (16 \times 2^{17} \times 115200 / 12 \times 10^6) = 20132.65_d$

$\approx 20133_d$ (in decimal)

$\approx 4EA5_h$ (in hexadecimal)

= 01001110 & 10100101 = n<15, 14, 13,.....1, 0>

As can be observed, n<15, 14, 13,.....1, 0> need external signals of "01001110" & "10100101" to program the baud rate generator through pins n<0> to n<15>. Since pin n<11> to n<15> are located to the parallel port, they can use the GXSPORT. However, the remaining pins, which not located to the parallel port will need an external circuit to drive their input pins.

6.5.3 External Circuit

Pin 2 (VCC) and 52 (GND) are special purpose pins that will be used to apply logic signals to the design. However, a safety precaution has to be made to prevent the possibility of shorting VCC and GND together. Shorting the VCC and GND together may damage the board used in this test. Each input pin has to be connected to jumper wires and 1kΩ resistors before applying it to pin 2 (VCC) or pin 52 (GND). This is to assure the current sink and current source of the XC4005E FPGA not to exceed 5mA (where: 5mA = 5V/1kΩ, 5V is the regulated power supply). The current sink is the current to the FPGA and the current source is the current from the FPGA. These currents may vary depending on the FPGA used. **Figure 6.9** shows how input signals can be applied to the FPGA pin using a switch at VCC.

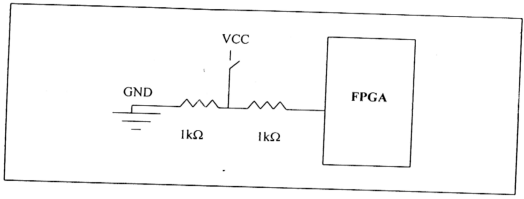
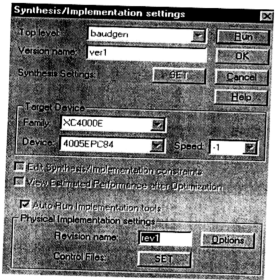


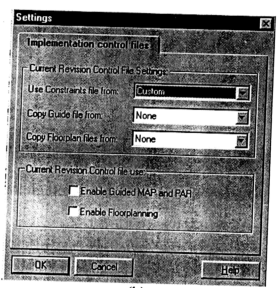
Figure 6.9: External circuit to drive inputs to FPGA pins

6.5.4 Download Bit Stream File

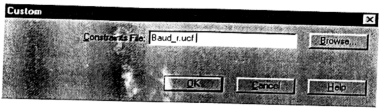
To produce a bit stream (**.bit) file that will follow the user-constraint files (**.ucf), the XSE2i implementation tools need to be directed to the constraint file.



(a)



(b)



(c)

Figure 6.10: Custom Implementation

Figure 6.10 shows windows pane of XSE2i implementation tools that will be used to specify the pin location using user constraint file (**.ucf). The user has to select "SET" button in "Synthesis/Implementation Settings" (**Figure 6.10(a)**) pane to invoke the "Settings" pane (**Figure 6.10(b)**). At the "Settings" pane, the "Use Constraint File" is selected as custom. The selection will invoke "Custom" pane (**Figure 6.10(c)**) and then the user constraint file (**.ucf) is selected by using the "Browse" button. Following the selection, both "Settings" and "Custom" panes are closed by using "OK" button. To start the implementation, the "Run" button is selected at "Synthesis/Implementation Settings" pane. After the implementation has been completed, the bit stream (baud_r.bit) file is then selected in a revision folder (rev1). The revision folder is located inside the folder that has been used for creating the VHDL project. To program the XS40 prototyping board, the bit file produced by XSE2i implementation tools is selected and dragged into the GXSLOAD.

6.5.5 Baud Rate Test Result

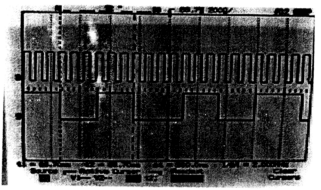


Figure 6.11: Baud Rate Test Result

Figure 6.11 shows two output waveforms of baud rate generator selected at pins 13 and 68 using mixed signal oscilloscope. The first (top) waveform is the hclk produced by 12 MHz-clock resonator at pin 13 and the second waveform shows the 16X clock for driving the internal transmitter logic (bclkX16) at pin 68. The oscilloscope also shows that the frequency for hclk is ranged from 11.9 MHz to 12.5 MHz and the bclkX16 is ranged from 1.701 MHz to 2 MHz.

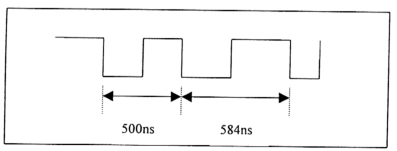


Figure 6.12: Measurement of BclkX16 using Mixed Signal Oscilloscope

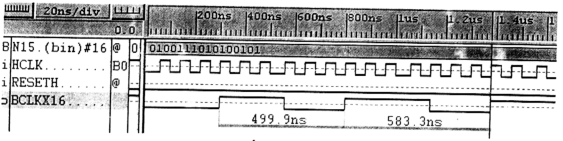


Figure 6.13: Simulated waveform of BclkX16

A comparison between **Figure 6.12** and **Figure 6.13** show small differences between the simulated waveform and the real waveform. The difference of the captured waveform and the timing simulation waveform is less than 1ns at that point. If the generated waveform is consistent, the data will transmit at ≈ 115.2 kbps (the transmission rate). The calculation of the transmission rate is as follows:

[(500 X 8) + (584 X 8)] ns = 8.672 ms

1/8.672ms = 115313.65 bps ≈ 115.2 kbps

6.6 BILBO Test

The bilbotst.bit is a bitstream file which has been constrained with bilbotst.ucf. The pad report file in **Appendix 3** reports the pin location of the implemented "BILBO Test" design. This will guide the user to select or apply logic signals to the inputs and observe the outputs. The pad report is located at "Reports => Implementation Report Files => Pad Report" in XSE2i Project Manager. Bilbo_mode<0> (pin 45) and Bilbo_mode<1> (pin 46) are located at the PC parallel data output (PC_D1 and PC_D2). The programmed FPGA will be configured as its operating mode (**Table 6.2**) according to PC_D2 & PC_D1, which will be set from GXSPORT. In summary the BILBO operating modes are as follows:

Table 6.2: BILBO Operating Mode

PC_D2 & PC_D1	Operating Mode
00	Shift register
01	PRPG
10	Normal
11	MISR

6.6.1 Pseudo Random Pattern Generation (PRPG)

To feed a "seed" value for initializing the LFSR, the programmed FPGA (BILBO) will need to be configured as a shift register ("PC_D2 & PC_D1" = "00"). The

shift register will shift the value of "si" (pin 48, PC_D4). In this test, "si" is set to '1' (PC_D4 = '1') to initialize the LFSR to "11111111" (i.e. "00000000", "00000001", "00000011", "00000111",..... "01111111", "11111111", 'si'='1' shifts from LSB to MSB each clock cycle). For further test, the LFSR can be initialized by changing the "si" states using a programmable signal generator and its appropriate software.

After the LFSR has been initialized, the programmed FPGA (BILBO) needs to be set to PRPG mode ("PC_D2 & PC_D1" = "01") to generate pseudo random pattern signals. The result of the pseudo random pattern generator (PRPG) waveforms will be observed using mixed signal oscilloscope.

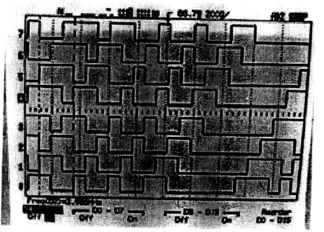


Figure 6.14: Pseudo Random Pattern

The left most data on **Figure 6.14** (the dotted line) is observed as "00111011" (MSB....LSB) followed by "01110110". The waveforms obtained have proven the result of 8-bits PRPG in simulation and theory. The PRPG is achieved by XOR-ing bit 1,2,3 and 7 (MSB...LSB, b7...b0) then the XORed result is then placed to bit 0 (LSB). The

other remaining bits (b6…b0) are then shifted to the left. Therefore, the result will be "01110110". Below shows how the LSB is achieved:

'1' (bit 1) ***XOR*** **'0'** (bit 2) ***XOR*** **'1'** (bit 3) ***XOR*** **'0'** (bit 7)= **'0'** (LSB)

6.6.2 Multiple Input Signature Register (MISR)

The test setup for MISR is almost the same as the PRPG except that the operation mode needs to be set to MISR ("PC_D2 & PC_D1" = "11") after initialized with "seed" value (using "si" and shift register). For this test, z<7, 6, 5,…..1, 0> are set to "00000111" using an external circuit. These data will act as the data output of the circuit under test. The MISR outputs are then observed at q<7, 6, 5, ….. 1, 0> (pin 65, 60, 61, 62, 59, 58, 50, 57) using mixed signal oscilloscope.

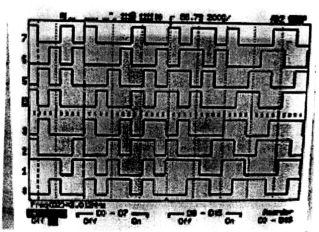


Figure 6.15: Multiple Input Signature Register (MISR)

Figure 6.15 shows the MISR result observed with mixed signal oscilloscope. The left most data (the dotted line) is observed as "00101010" followed by "01010011". The MISR is achieved in a similar procedure as the PRPG but the produced data (PRPG) is then XORed with the output data of the circuit under test (z<7, 6, 5, ... 1, 0>). Below shows how the signal result is produced.

"00101010" XORed and shift will produce "01010100"

"01010100" XORed with "00000111" will produce "01010011"

6.7 UART Internal Loop Back Mode

UART internal loop back mode will be used to test the transmitter and the receiver of the designed UART. In this mode, parallel data at data<7,6,5,....1,0> (pin 77, 70, 69, 66, 9, 8, 7, and 6) will be transmitted in serial. The serial data then will be looped back to the receiver, which converts the serial data back to parallel data. The parallel data received at the receiver can be observed at the same pin as the transmitter's parallel data (data<7,6,5,....1,0>) when data output is enabled. This is possible since the pins have been specified as bi-directional in the VHDL description. However, to avoid confusion between the input data and the output data, some modification are made to the design. In this test, eight pins will be reserved for received parallel data (rxdatest<7, 6, 5,.....1, 0>) at pin 65, 60, 61, 62, 59, 58, 50, 57. (Note: Please refer to **Appendix 4** and **Appendix 5** for the user constraint file and the pad report file, which show the pin locations of the UART with BIST).

To start the test, external circuit with switch will be constructed to drive logic signals to data<7, 6, 5,....1, 0>. The data bus (data<7, 6, 5,....1, 0>) acts as a communicator between the UART and the CPU. The reset (reseth => PC_D0), address (addr<2, 1, 0> => PC_D1, PC_D2, PC_D3), read/write (rw => PC_D4) and chip select (cs => PC_D5) are located to the PC parallel data outputs using "hwtst.ucf". The pins are located to the PC parallel port to provide them with the input signals from GXSPORT. Because of insufficient place at GXSPORT, address3 (addr<3>) will use external circuit with switch at pin 24. Bilbo_mode<1>, ldareg, ldbreg, rxd are set to '1' and the other remaining input pins are set to '0' (using external circuit).

After the circuit and pin location setup is complete, the "hwtst.bit" is downloaded to XS40 using GXSLOAD. In this section, the test will be carried out as the same procedure as the simulation done in **Chapter 5**. The process is proceed as follows:

Table 6.3: Test Procedure

	PC Parallel data outputs							Data<7, 6, 5,...1, 0> (pin 77, 70, 69, 66, 9, 8, 7, and 6)
	Cs	Rw	Reseth	Addr<3, 2, 1, 0>				
No.	PC_D5	PC_D4	PC_D3	<3> pin 24	<2> PC_D2	<1> PC_D1	<0> PC_D0	
1	1	0	1	0	0	0	0	"00000000"
2	1	0	0	0	0	0	0	"00000000"
3	0	0	0	0	0	0	0	"00000000"
4	1	0	0	0	0	0	0	"00000000"
5	1	0	0	0	0	0	1	"00000111"
6	0	0	0	0	0	0	1	"00000111"
7	1	0	0	0	0	0	1	"00000111"
8	1	0	0	0	0	1	0	"00001111"
9	0	0	0	0	0	1	0	"00001111"

10	1	0	0	0	0	1	0	"00001111"
11	1	0	0	0	0	1	1	"10100101"
12	0	0	0	0	0	1	1	"10100101"
13	1	0	0	0	0	1	1	"10100101"
14	1	0	0	0	1	0	0	"01001110"
15	0	0	0	0	1	0	0	"01001110"
16	1	0	0	0	1	0	0	"01001110"
17	1	0	0	1	0	0	0	"00000111"
18	0	0	0	1	0	0	0	"00000111"
19	1	0	0	1	0	0	0	"00000111"
20	1	1	0	1	0	0	0	"00000111"
21	0	1	0	1	0	0	0	"00000111"
22	1	1	0	1	0	0	0	"00000111"
23	1	0	0	1	0	0	0	"01111111"
24	0	1	0	1	0	0	0	"01111111"

Table 6.3 shows the programmed FPGA (UART with BIST) is reset using PC_D3. This is to make sure that the UART will be initialized before the test can be conducted. Next, reseth is set to '0' to begin the test. Cs is then set to '0' to choose the data and address to program the UART. At this point, ack (pin 68) will be activated; acknowledging the selected data has been received. The shaded columns (i.e. 3, 6, 9...24) show the selected data at corresponding address (ADDR[3:0]). **Table 6.4** summarized the description of the selected addresses and data.

Table 6.4: Summary of the selected ADDR[3:0] and DATA[7:0]

Addr <3,2, 1, 0>	**Data <7, 6, 5,...1, 0>**	**Description**
"0000"	"00000000"	Addr = "0000" selects LCR1 Data = "00000000" • 8 bit mode • 1 stop bit • no parity bit • disable stick parity mode • disable break control

"0001"	"00000111"	Addr = "0001" selects LCR2 Data = "00000111" • enable transmitter • enable receiver • internal loop mode • disable RTS • disable CTS
"0010"	"00001111"	Addr = "0010" selects MCR Data = "00001111" • Force DTR to '1' • Force RTS to '1' • Force out1 to '1' • Force out2 to '1'
"0011"	"10100101"	Addr = "0011" selects BRSEL0 Data = "10100101" • BRSEL0 = "10100101" (to transmit at 115.2 kbps with 12 MHz clock)
"0100"	"01001110"	Addr = "0100" selects BRSEL1 Data = "10100101" • BRSEL1 = "01001110" (to transmit at 115.2 kbps with 12 MHz clock)
"1000"	"00000111"	Addr = "0100" selects holding register Data = "0000111" • Data = "0000111" is transmitted

At Addr = "1000" and Data = "00000111" the received data is observed at rxdatest<7, 6, 5,.....1, 0> (pin 65, 60, 61, 62, 59, 58, 50, 57) using mixed signal oscilloscope. The oscilloscope shows that rxdatest<7, 6, 5,.....1, 0> = "00000111", which is equal to the data transmitted when Addr in holding register mode (Addr = "1000"). The data ("0000111") will not change even there is a new data that should be transmitted at the UART's transmitter (because the buffer register is full). To clear the buffer, rw must be set to '1' (No. 20) forcing the CPU to read the data in the UART's register. Data = "01111111" is then transmitted. The result of the test can be observed in

Figure 6.16 where '9' (top waveform) is the waveform of ack, '8' is the waveform of 12 MHz clock, and '0' to '7' are the rxdatest<7, 6, 5,.....1, 0>.

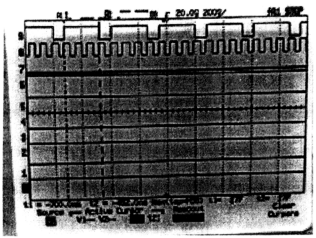


Figure 6.16: UART Internal Loop Test Result

The hardware test has managed to show the reliability of the designed UART chip to work well in real world application. The test has shown that the outcomes of the hardware test are similar to the theory and simulation. In the next chapter, a discussion on the hardware overhead when implementing BIST to the UART design will be presented. The discussion then will be followed by the conclusion of the thesis.