I.J. Information Engineering and Electronic Business, 2015, 6, 23-27
Published Online November 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2015.06.04

Single Chip Embedded System Solution: Efficient Resource Utilization by Interfacing LCD through Softcore Processor in Xilinx FPGA

Shiraz Afzal, Farrukh hafeez and Muhammad Ovais Akhter
Department of Electronic Engineering, Sir Syed University of Engineering and Technology, Karachi-Pakistan 75300
Department of Electronic and Electrical Engineering Jubail Industrial College, Jubail- Saudiarabia
Department of Electronic Engineering, Sir Syed University of Engineering and Technology, Karachi-Pakistan 75300
Email: safzal@ssuet.edu.pk,f_hafeez@jic.edu.sa, moakhter@ssuet.edu.pk

Abstract—This paper presents a customized application in term of LCD interfacing through soft-core microcontroller by implementing its control circuitry in FPGA. Therefore optimum efficiency is achieved. This provide the single chip solution by replacing conventional model which contain separate microcontroller circuit, therefore the efficient resource utilization of FPGA is made, day by day as an increase in the complexity of Embedded system design thus gives many advantages over conventional designed processors such as platform independence, less cost, flexibility and greater immunity to obsolescence.

Index Terms—FPGA, LCD interfacing, soft-core microcontroller e.t.c.

I. Introduction

A microcontroller is the major part of any embedded system that uses both hardware and software components to perform specific application. As embedded systems play a vital role in our everyday life also exist in profusion in our modern civilization and can be found in places such as in our vehicle, in health field, in industries and in amusement electronics to name just a few [1].

As an increase in complexity of embedded system. It became very difficult to design each hardware component because of the cost and become impractical approach therefore the emergence of Soft core microcontroller solve this problem in which a designer can easily build their design and verify that.

Here in this manuscript an important hardware element that relates to the embedded system which is LCD interfacing through soft-core controller is being controlled through Xilinx picoblaze controller as shown in Fig. 1.

The material of the manuscript is arranged as follow in section II explains the survey of softcore processors. Section III we describe the software used for HDL Language. Section IV describe the FPGA implementation which involves subsections as follow: Sub-section "a" describe the Pico-blaze Microcontroller Overview. Sub-section "b" describes the VHDL component and design Processes. Sub-section "c" describes the picoblaze instruction set and sub-section "d" describes the LCD display Overview.

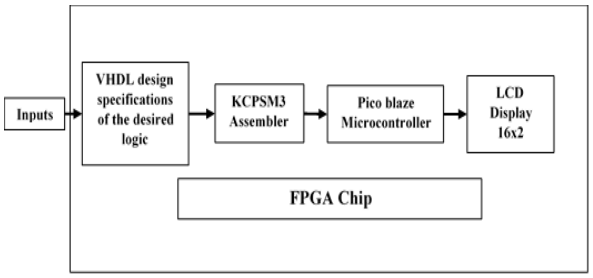

Fig.1. LCD interfacing through soft core processor

II. Review of Available Soft-Core Processors

This segment provide a list of available softcore processor specified by the both open source communities and commercial vendorsare provided

A. Open Resource Softcores

An open source community provides various cores freely available as given in [2]. These types of softcores are not only used for research purpose but also for the development of embedded systems [3], an example of such type of core used in academia for research purpose is UT Nios [4]. Another listof cores which is not only used forthe ASICs but can be also used for modeling and synthesize of FPGA is OpenSPARC [5]. Further LEON [6], LEON2 [6], LEON3 [5], Open RISC 1200 [2] and sun microsystem as provides theirown core are also available as open source soft cores. Details of which are available in [3].

B. Commercial Available Softcores:The major softcore controller includes Microblaze, Picoblaze by Xilinx, Nios II by Altera and Xtensa by Tensilica [3].

Microblaze and Picoblaze: This 32bit and 8bit softcore processor are incorporated by Xilinx for both Spartan and Virtex series, one of the leading vendor of FPGA. Microblaze is one of the demanding and mostly used

softcore processor. It uses a Harvard RISC architecture, having 32 bit instruction set, and two level of interrupts and can operate up to 200MHz on Virtex 4 FPGAwhile Picoblaze is a compact 8 bit softcore controller can be useful for simple data processing application because of its miniature element and cost effective features [3].

Nios II by Altera: is a general purpose softcore processor that uses Harvard RISC architecture, 32 bit instruction set. It comes in three versions fast core, standard and economy further each core varies in performance and dimension depending on the chosen feature

III. Tool Used

Xilinx 8.2i is used for the programming and configuring of FPGA, It give user friendly environment and can also provide flexibility in terms of by either implementing the logic using basic logical operator or with schematic diagram using schematic editor implementation.

Here in this task the LCD is programmed using VHDL language, further for picoblaze softcore processor KCPSM3 assembler is used. DOSBOX 0.74 is used for picoblaze psm file simulation.

IV. Implementation

The work is carried out in the following segments:

A.Picoblaze Processor overview.
B. VHDL Components and design processes.
C.Picoblaze instruction set.
D. LCD display overview.

A. Picoblaze Overview

Fig. 2 shows the block diagram of picoblaze microcontroller, the significance of using picoblaze is of its small size, having RISC architecture and less FPGA slice available only 96 slices. It is given as a free, source-level VHDL file with royalty-free re-use within Xilinx FPGAs [8].

There main feature includes 64 bytes of internal memory, 256 input and output ports available, 16 byte wide general purpose register.

B. VHDL Component and Design Processes

VHDL components used by the picoblaze controller is shown in Fig. 3.They are mainly divide into two components that is block memory ROM and KCPSM3, KCPSM3 component provide the registers, Arithmetic logic unit etc.

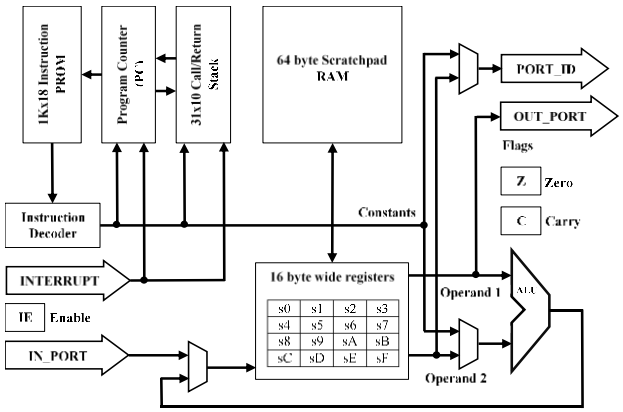

Fig.2. Block diagram of Picoblaze microcontroller

C. PicoblazeInstruction Set: The instructions mainly used by the picoblaze microcontroller is shown in table [3 manualref.].

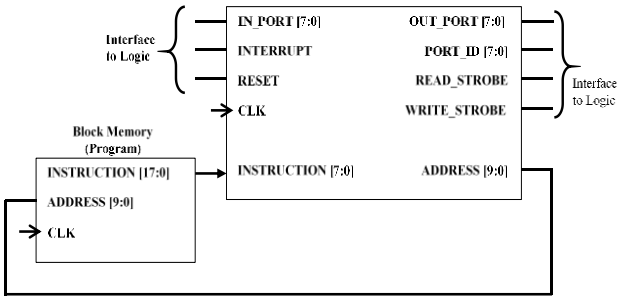

Fig.3. VHDL components used by the picoblaze controller

Table 1. VHDL Instructions with Description

Instruction	Description
ADD aX, mm	Add register aX with literal mm.
ADD aX, bY	Add register aX with register bY.
ADDCY aX, mm	Add register aX with literal mm with CARRY bit.
ADDCY aX, bY	Add register aX with register bY with CARRY bit.
AND aX, mm	Bitwise AND register aX with literal mm.
AND aX, bY	Bitwise AND register aX with register bY.
CALL zzz	Unconditionally call subroutine at zzz.
CALL C, zzz	If CARRY flag set, call subroutine at zzz.
CALL NC, zzz	If CARRY flag not set, call subroutine at zzz.
CALL NZ, zzz	If ZERO flag not set, call subroutine at zzz.
CALL Z, zzz	If ZERO flag set, call subroutine at zzz.
COMPARE aX, mm	Compare register aX with literal mm Set CARRY and ZERO flags asappropriate. Registers are unaffected.
COMPARE aX, bY	Compare register aX with register bY. Set CARRY and ZERO flags as appropriate. Registers are unaffected
DISABLE INTERRUPT	Disable interrupts.
ENABLE INTERRUPT	Enable interrupts.
FETCH aX, bY	Read scratchpad RAM location pointed to by register bY into register aX.
FETCH aX, ss	Read scratchpad RAM location ss into register aX.
INPUT aX, bY	Read value on input port pointed to by register bY into register aX.
INPUT aX, pp	Read value on input port pp into register aX.
JUMP zzz	Unconditionally jump to zzz.
JUMP C, zzz	If CARRY flag set, jump to aa.
JUMP NC, zzz	If CARRY flag not set, jump to aa.
JUMP NZ, zzz	If ZERO flag not set, jump to aa.
JUMP Z, zzz	If ZERO flag set, jump to aa.
LOAD aX, mm	Load register aX with literal mm.
LOAD aX, bY	Load register aX with register bY.
OR aX, mm	Bitwise OR register aX with literal mm.
XOR aX, bY	Bitwise XOR register aX with register bY.
XOR aX, mm	Bitwise XOR register aX with literal mm.
TEST aX, bY	Test bits in register aX against register bY. Update CARRY and ZEROflags. Registers are unaffected.
TEST aX, mm	Test bits in register aX against literal mm. Update CARRY and ZEROflags. Registers are unaffected.
SUBCY aX, bY	Subtract register bY from register aX with CARRY (borrow).
SUBCY aX, mm	Subtract literal mm from literal aX with CARRY (borrow).
SUB aX, bY	Subtract register bY from register aX.
SUB aX, mm	Subtract literal mm from literal aX.
STORE aX, ss	Write register aX to scratchpad RAM location ss.
STORE aX, bY	Write register aX to scratchpad RAM location pointed to by register bY.
SRX aX	Shift register aX right. Bit aX[7] is unaffected.
SRA aX	Shift register aX right through all bits, including CARRY.
SR1 aX	Shift register aX right, one fill.
SR0 aX	Shift register aX right, zero fill.
SLX aX	Shift register aX left. Bit aX[0] is unaffected.
SLA aX	Shift register aX left through all bits, including CARRY.
SL1 aX	Shift register aX left, one fill.
SL0 aX	Shift register aX left, zero fill.
RR aX	Rotate aX register right.
RL aX	Rotate aX register left.
RETURNI ENABLE	Return from interrupt service routine. Re-enable interrupt.
RETURNI DISABLE	Return from interrupt service routine. Interrupt remains disabled.
RETURN Z	If ZERO flag set, return from subroutine.
RETURN NZ	If ZERO flag not set, return from subroutine.
RETURN NC	If CARRY flag not set, return from subroutine
RETURN C	If CARRY flag set, return from subroutine.
RETURN	Return unconditionally from subroutine.
OUTPUT aX, pp	Write register aX to output port location pp.

D. LCD Display

Spartan 3E uses 2×16 display consist of 2lines of 16 character each. LCD is controlled by FPGA through an interface and also connected with Intel Startaflash whose function is to give full write/read access to LCD as shown in the Fig. 4.

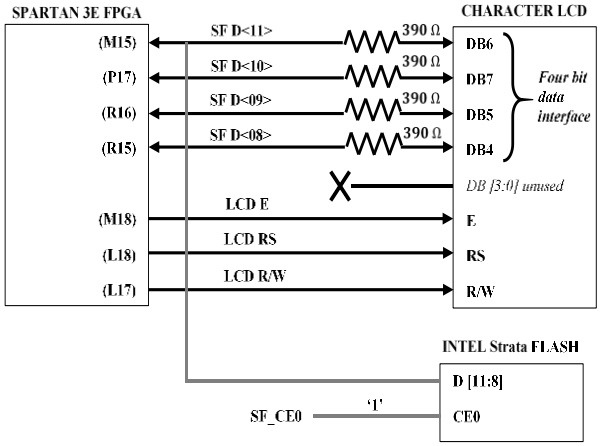

Fig.4. Read Write access to LCD

While the LCD display data ram locations shows in Fig. 5, only 32 characters can be displayed at a time.

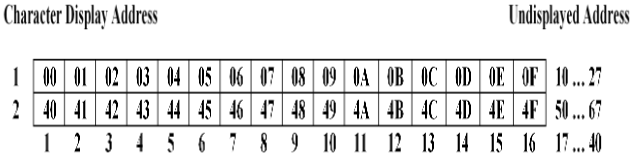

Fig.5. LCD Display

The commands to controlled LCD is sum up in the Table 2.

Table 2. Commands Controlled LCD

Function	LCD_RS	LCD_RW	Upper Nibble				Lower Nibble			
			DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear Display	0	0	0	0	0	0	0	0	0	1
Return Cursor Home	0	0	0	0	0	0	0	0	1	-
Enter Mode set	0	0	0	0	0	0	0	1	ID	S
Display On/Off	0	0	0	0	0	0	1	D	C	B
cursor and Display Shift	0	0	0	0	0	1	SC	RL	-	-
Function Set	0	0	0	0	1	0	1	0	-	-
Set CGRAM Address	0	0	0	1	A5	A4	A3	A2	A1	A0
Set DDRAM Address	0	0	1	A6	A5	A4	A3	A2	A1	A0
Read Busy Flag and Address	0	1	BF	A6	A5	A4	A3	A2	A1	A0
Write Data to CGRAM or DDRAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0
Read Data from CGRAM or DDRAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0

The font bitmap for the characters stored in character generator ROM, It contains the Japanese kana characters and ASCII characters as shown in the Fig. 6.

In this assignment a KCPSM3 assembler is used which translates the input psm file and also three block RAM initialization templates and produces the output as shown in Fig. 7.

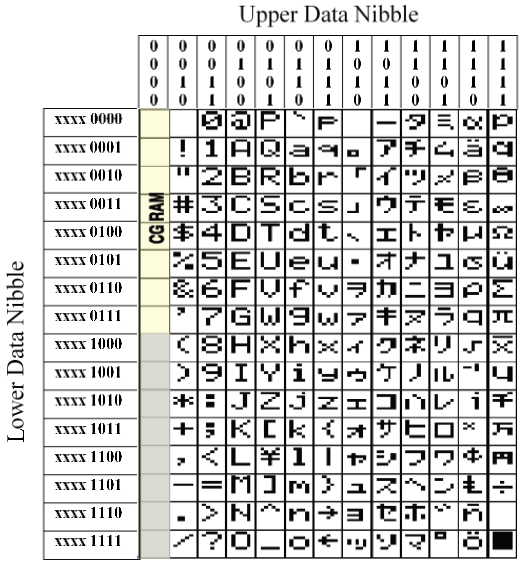

Fig.6. Font Bitmap Characters for LCD Display

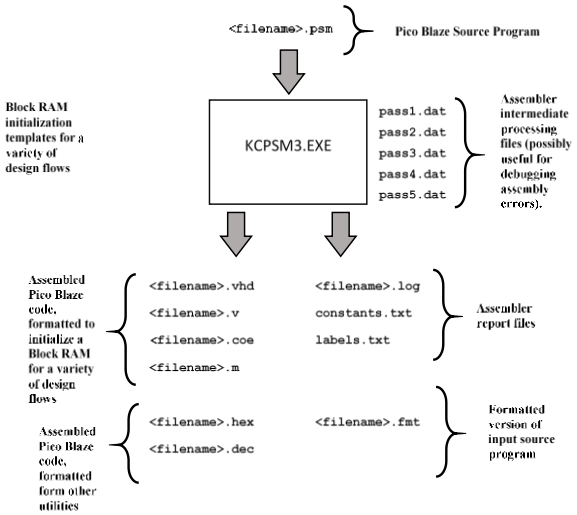

Fig.7. KCPSM3 Assembler

V. Simulation Results

Fig. 9 shows the simulation result by displaying the name on the 2 line 16 character LCD using softcore picoblaze controller of Xilinx Spartan 3E FPGA.

Fig.9. Result display on LCD

REFERENCES

[1] Mishra, P. and Dutt, N., "Architectural description languages for programmableembedded system",IEE Proceedings of computer and digital technique, May 2005, pp. 285-297.

[2] Opencores.org Website, www.opencores.org, June 2006.

[3] Jason GT. Tong, Ian D. L Anderson and Muhammad A. S. Khalid, "Softcore processors for embedded system", IEEE Microelectronics International Conference, Dec 2006, pp. 170-173.

[4] UT Nios Homepage, www.eecg.toronto.edu/Uplavec/utnios.html, June, 2006.

[5] Open SPARC Website, www.opensparc.org, June 2006.

[6] "GRLIB IP Core User's Manual", Gaisler Research, February 2006.

[7] "LEON2 Processor User's Manual XST Edition", Gaisler Research, July 2005.

[8] PicoBlaze 8-bit Embedded Microcontroller User Guide, Online: http://www.xilinx.com/support/documentation/ip_documentation/ug129.pdf, accessed 3 Dec 2012.

Author's Profiles

Shiraz Afzal is working as a Assistant Professor in Electronic Engineering Department at SSUET. He already published nine research articles in various International Journals. He has 8 years of teaching experience. He received his B.E. degree in Electronics from Sir Syed University of Engineering and Technology Karachi, Pakistan and M.E degree in Electronics, specialization in Micro-System design from NED University of Engineering & Technology Karachi, Pakistan in 2006 and 2012 respectively. His research interest includes Microelectronic circuit design and Embedded System Design. He is a member of PEC.

Farrukh Hafeez is working as a lecturer in Electrical and Electronic Engineering department at Jubail Industrial College. He received his B.E. degree in Electronics from NED University of Engineering and technology Karachi, Pakistan and M.E degree in Electronics, specialization in Industrial control system from NED University of Engineering & Technology Karachi, Pakistan in 2005 and 2009 respectively. His research interest control system design and applications.

Muhammad Ovais Akhter is Assistant Professor in SSUET. Graduated in Electronic Engineering from same university. Over 6 years of experience in the area of Modeling, Simulation and Signal Processing.

How to cite this paper: Shiraz Afzal, Farrukh hafeez, Muhammad Ovais Akhter,"Single Chip Embedded System Solution: Efficient Resource Utilization by Interfacing LCD through Softcore Processor in Xilinx FPGA", IJIEEB, vol.7, no.6, pp.23-27, 2015. DOI: 10.5815/ijieeb.2015.06.04