Software Defined 8, 16 & 24 Bit Digital Logic Design by One Microcontroller

khaled hossain [1] and Md. Nasimuzzaman Chowdhury[2]

[1] American International University Bangladesh

Received: 12 December 2012 Accepted: 5 January 2013 Published: 15 January 2013

Abstract

Now-a-days digital circuits are getting more complex. One IC in a digital circuit is used for a fixed purpose and its operation cannot be defined through software. Because of this limitation digital circuit becomes larger in size. When designing an 8bit digital circuit we do not include 16bit or 24bit components, but this limits our scope of design and versatility of the design. To overcome this problem an 8bit microcontroller is programmed which is able to do addition, subtraction, multiplication and 28 other digital operations in 8, 16 24 bit level. To add six 8 bit data 5 adder ICs are not needed anymore. This IC can do it all alone. For any logic operation the regarding mode needs to be selected in the same IC to perform desired operation. It is software defined digital logic design IC. This IC will save time, space reduce cost in digital circuit designing.

Index terms— microcontroller, digital logic IC, universal shift register, encoder, decoder.

1 Introduction

icrocontrollers are capable of executing all digital logic design operations. In another paper we have implemented 28 digital logic design operation by one microcontroller [1]. But still if a 16bit operation is needed we have to cascade two 8bit microcontrollers.

Instead of cascading two microcontrollers, it is possible to do operations up to 24bits with this IC. This IC can perform AND, OR, NOR, NAND, XOR, XNOR, Universal Shift Register, Counter operations etc. The Controller used in this project is Atmega1280. It has 86 GPIO. In this project there are 6sets of 8bit input, 25bit output, 6bit main operation selection option and additional 5bit to select sub operations. **??**hen Rest of the pins of portG represent sub operations and to select main operation portH is used. Enable_bar means to enable the IC we need to make portH6_bit low. PortH7 represent syncronous clock input. When 16bit operations are selected PortA are lower eight bits & PortB are higher eight bits. In the same way for input2 & input3 portF, Portk are lower eight bits and PortJ, PortL are higher eight bits accordingly. For 24bit operation in input1 portA are lower eight bits, portB are higher eight bits, portF are most significant eight bits. For second input PortJ are lowest significant bits, portK are higher eight bits, portL are highest eight bits. At output, portC are most significant 8bits, portD are higher 8bits, PortE are more higher 8bits & portG6_bit is the MSB.

2 Circuit Description

Atmega1280 is used as microcontroller in this project. Operating voltage of atmega1280 is +5v. Here VCC is +5v. Atmega1280 has 86 GPIO, 11 GPIO ports. They are PORTA, PORTB, PORTC, PORTD, PORTE, PORTF, PORTG, PORTH, PORTJ, PORTK and PORTL. Among these ports PORTA, PORTB, PORTF, PORTJ, PORTK, PORTL, PORTH and PORTG are all input. PORTC, PORTD, PORTE & PORTG5_bit are all output. To give input toggle switches are used and to see the output LEDs are used. There are two more inputs one is Enable and another is for clock. Microcontroller has its own clock connected to Xtal1 & Xtal2 pin. Value of oscillator is 16 MHz. A reset circuit is connected with RESET_bar pin to reset the microcontroller.

This microcontroller can execute 28 operations and lots of sub operations. To select main operation PORTH0 to PORTH5 total 6 pin is used and to select sub operation PORTG0 to PORTG4 is used. To Enable and disable the IC, Enable_bar is used at PORTH6_bit. External clock is used at PORTH7_bit. This IC can operate in both asynchronous and synchronous mode. When there is no external clock it will operate in asynchronous mode and while clock is used it will operate based on the clock frequency. If Enable_bar pin is high at that time microcontroller will not take any clock input or will not work in asynchronous mode also.

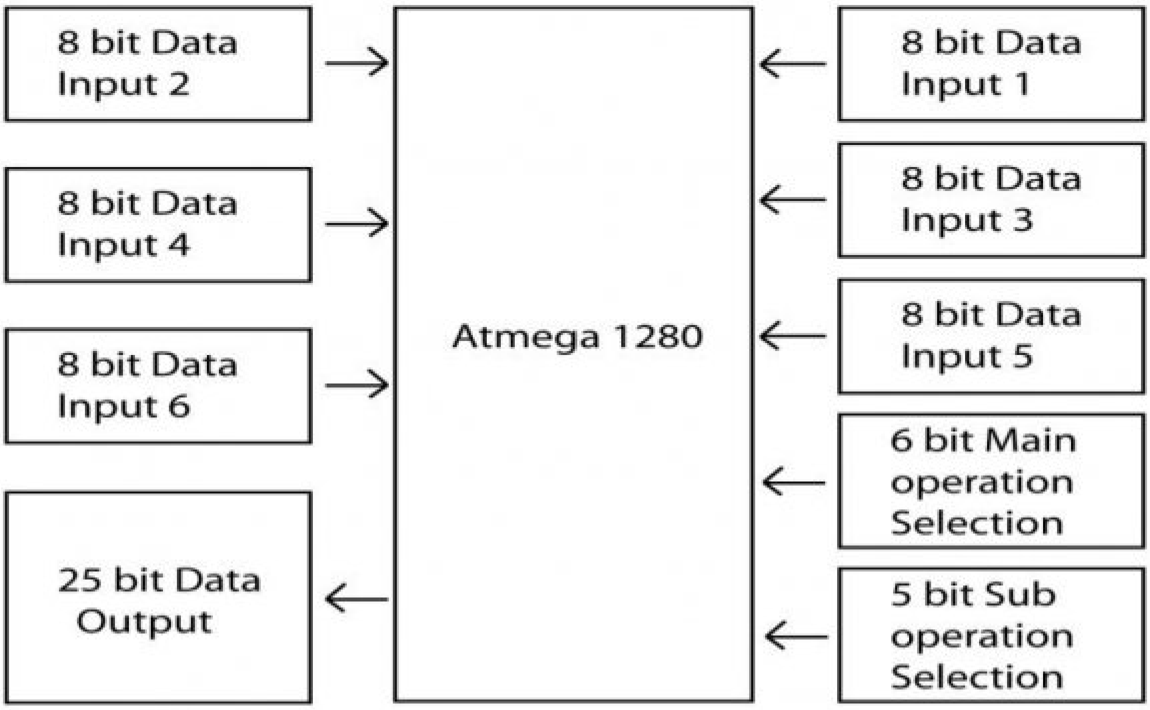
3 a) Main Technology Used

Main technology of this IC is the programming method. To perform any operation with this IC, at first the main operation mode and sub operation mode is selected. For example if we want to do an addition operation the mode 00 is selected. PortH0 to PortH5 has to be 0. Then we have to select sub operation like we want to do an 8bit, 16 bit or 24bit addition. If we want an 8bit addition we have to set the value of PortG0_bit & PORTG1_bit equal to 0. As this is an 8bit operation we have the option to choose with how many variable we would like to do the operation. We can choose an addition operation from two variables to six variables maximum. Let's choose 2 variables. Then Value of PortG2_bit is 0, PortG3_bit is 1 & PortG3_bit is 0. Here PortG2_bit is LSB and PortG3_bit is MSB. Now it is ready to perform two 8 bit additions. Any value of PortA will be added with any value of PortB and output will be shown by LEDs in binary format. PortC0_bit is the LSB bit of output and PortG5_bit is the MSB bit.

For an 8bit operation we need to select number of inputs but for 16bit or 24bit operation this IC will start working instantly when main operation and sub 24 bit which have sub operations also. For example if we want to execute a 24bit Barrel shifter, we have to set the main operation first. H5=0, H4=1, H3=1, H2=0, H1=1, H0=0. Then we have to select sub operation that, shift it right or shift it left. So, the sub operation G0=0, G1=0 is set for shifting right arithmetic. Then we have to select the amount of shifting through Pin G2 to G4. The table below gives us the operation chart and bit values to set them In the above table sub operation of addition is shown and how to select the operation bit values are also shown. In this way all the other operations are performed. If we wants to do subtraction IN1 has to be the largest number than second largest should be IN2, so that we don't get any wrong answer for subtraction. This order is maintained for division also. For two inputs, the equation is IN1/ IN2. Segment Decoder takes input as binary value and gives output for common cathode without dot, common cathode with dot, common anode without dot and common anode with dot. For example if L3=0, L2=1, L1=0, L0=0 then segment with show 4 as output. Ones counter, binary counter, Mux, Demux all these operations are also divided according to the above table. But Universal shifter and barrel shifter have other sub functions. These functions are shown below: G4 G3 G2 IN1 IN2 IN3 IN4 IN5 IN6 Equation Operation 0 1 0 PortA PortB X X X X IN1+IN2 ADD 0 1 1 PortA PortB PortF X X X IN1+IN2+IN3 1 0 0 PortA PortB PortF PortJ X X IN1+IN2+IN3+IN4 1 0 1 PortA PortB PortF PortJ Portk X IN1+IN2+IN3+IN4+IN5 1 1 0 PortA PortB PortF PortJ Portk PortL IN1+IN2+IN3+IN4+IN5+IN6L3 L2 L1 L0 OPERATIONS SUB X 0 0 0 Hold SIPO SISO PISO PIPO X 0 0 1 Load X 0 1 0 Load & Shift Right X 0 1 1 Load & Shift Left X 1 0 0 Shift Circular Right X 1 0 1 Shift Circular Left X 1 1 0 Shift Arith Right X 1 1 1 Shift Arith Left 0 0 0 0 Shift 0 Barrel Shifter 0 0 0 1 Shift 1 0 0 1 0 Shift 2 0 0 1 1 Shift 3 0 1 0 0 Shift 4 0 1 0 1 Shift 5 0 1 1 0 Shift 6 0 1 1 1 Shift 7 ? ? ? ? ???. 1 1 1 1 Shift 15

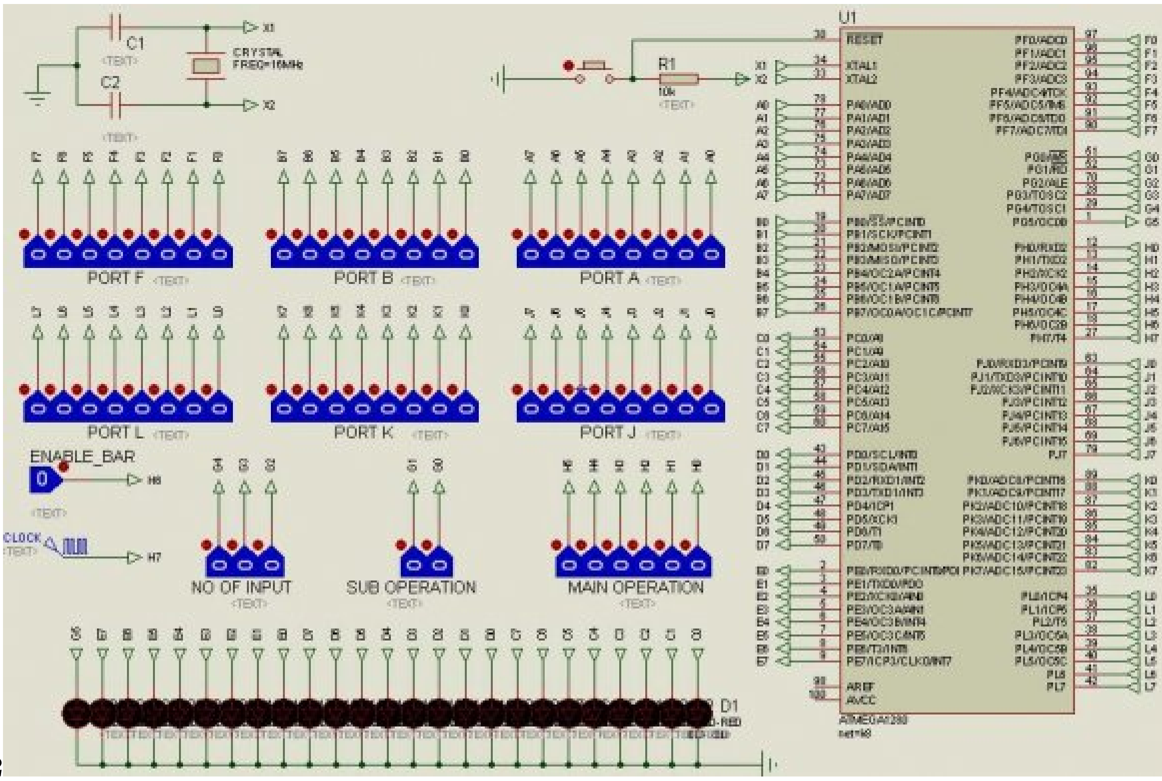
Atmega1280 contains 135 powerful instructions like addition, subtraction, shifting etc and most single clock cycle execution.32x8 general purpose working resistors, two 8bit timer & four 16bit timer, interrupt and wakeup on pin change. It comes in 100 lead TQFP, 100 Ball CBGA wich can be breakout easily and small in size. This microcontroller has the most number of GPIO and all 86 GPIO pins can be used as input or output. Brown out feature of this Ic makes it more stable output and long lasting. [1]

Figure 1:



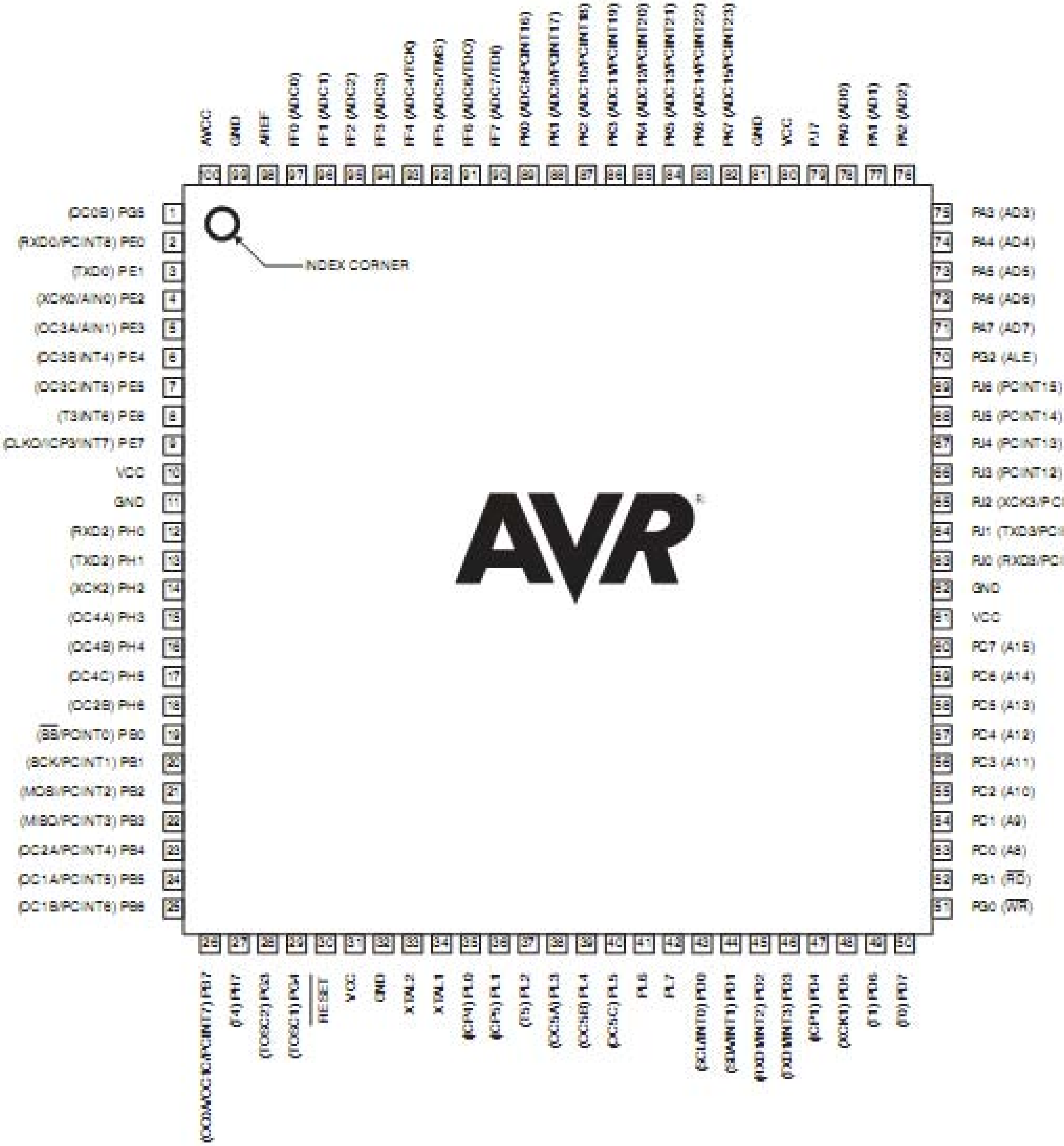
1

Figure 2: Figure 1 :



2

Figure 3: Figure 2 :



3

Figure 4: Figure 3 :

4

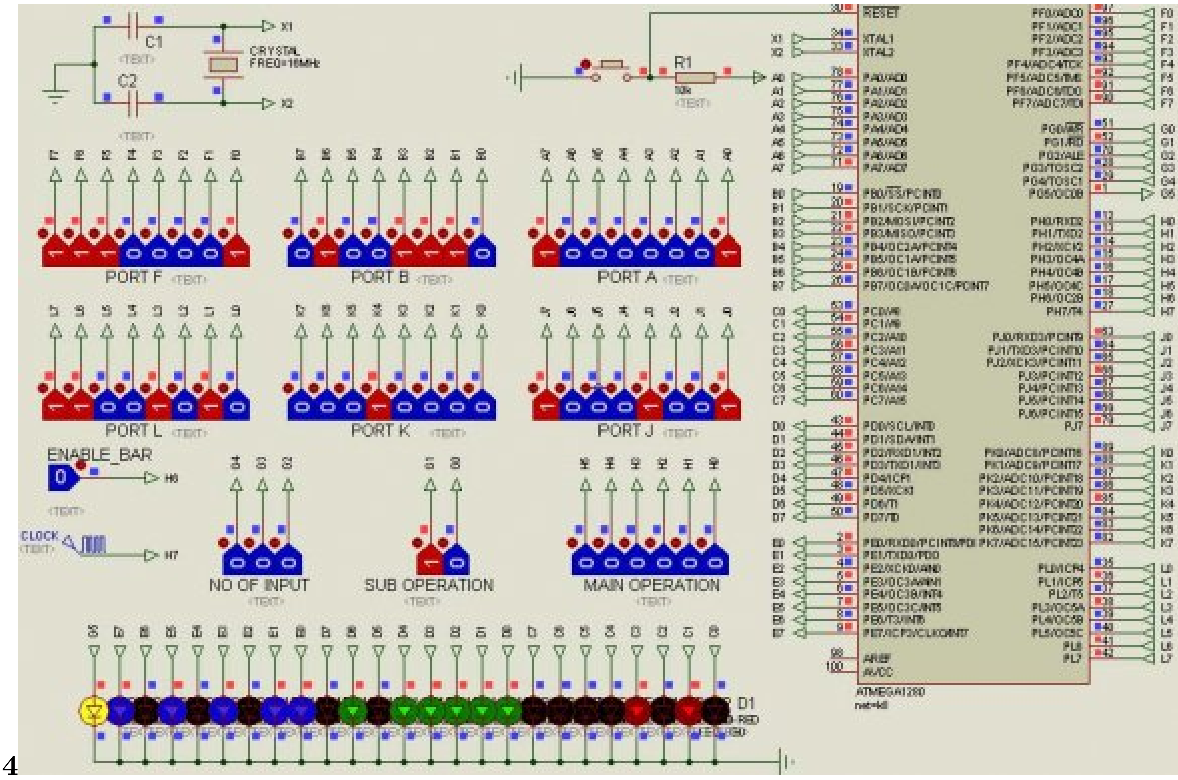


Figure 5: Figure 4 :

5

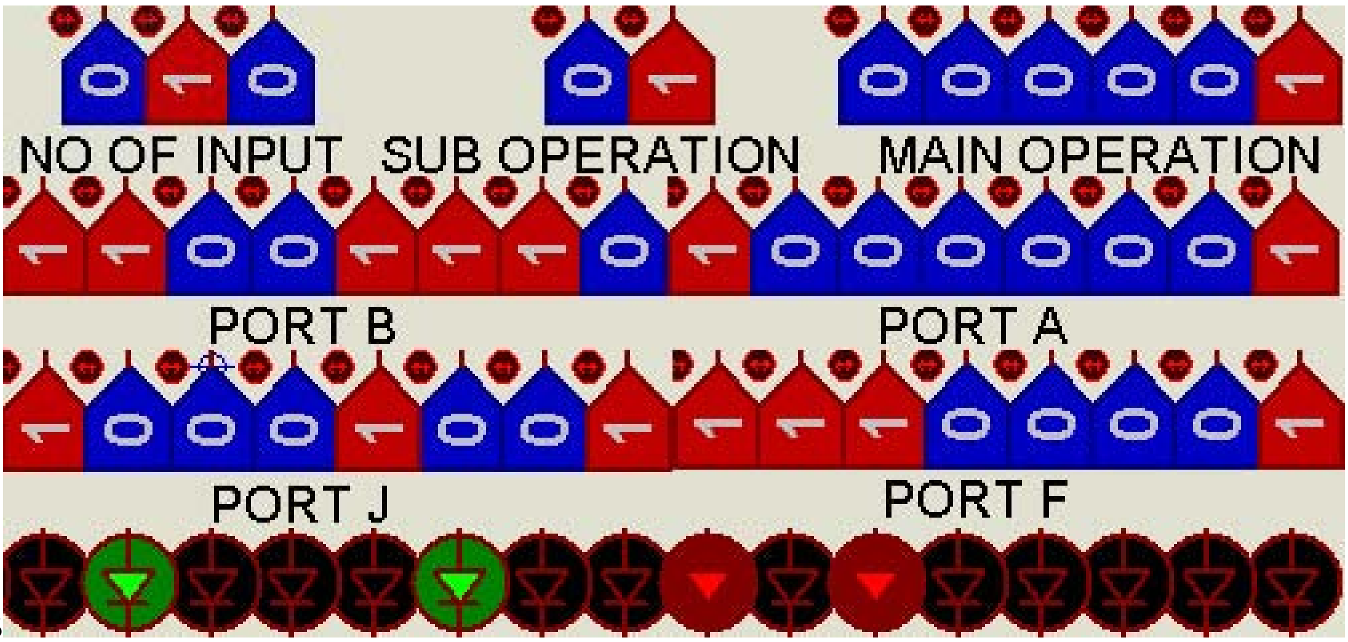


Figure 6: Figure 5 :?

1

Main Operation Chart

	H4	H3	H2	H1	H0	Operation
00	0	0	0	0	0	Addition
01	0	0	0	0	1	Subtraction
02	0	0	0	1	0	Multiplication
03	0	0	0	1	1	Division
04	0 0	0	1	0 0	0	AND OR
05		0	1		1	
06	0 0	0	1	1 1	0	NOT NOR
07		0	1		1	
08	0 0	1	0	0 0	0	NAND XOR
09	0	1	0	1	1	XNOR
10		1	0		0	
11	0	1	0	1	1	Buffer
12	0	1	1	0	0	Universal Shift Register
13	0 0	1	1	0 1	1	Ones Counter
14		1	1		0	Binary Counter
15	0	1	1	1	1	Inverted Counter
16	1	0	0	0	0	Mux
17	1	0	0	0	1	D S
18	1	0	0	1	0	C C S
19	1	0	0	1	1	C
20	1	0	1	0	0	S C
21	1	0	1	0	1	S
22	1	0	1	1	0	F
23	1	0	1	1	1	F
24	1	1	0	0	0	F
25	1	1	0	0	1	C
26	1	1	0	1	0	E
27	1	1	0	1	1	F
28	1	1	1	0	0	J C

Table 2 : Sub operation chart

G1	G0	0 0 0 1	Sub Operation 8 bit 16 bit	Main Operatio
1	0		24 bit	All operations

Table 3 : Sub operation chart

G1	G0	Sub Operation 0 0	7 Segment	Main Operatio
0	1		14 Segment	
1	0		16 Segment	

Figure 7: Table 1 :

4

[Note: 8 bit level operations]

Figure 8: Table 4 :

5

: 16 bit level operations

Figure 9: Table 5

6

: 24 bit level operations

Figure 10: Table 6

7

Figure 11: Table 7 :

8

b) Device Used
Brain of this project is Atmega1280 micro-controller. It is an 8 bit Micro controller with RISC architecture. Its speed is up to 16 MIPS throughout at 16MHz. It has 128K bytes of flash and 4Kbytes EEPROM. Operating voltage 2.7v -5.5v, in active mode it consumes only 1.1mA & in sleep mode it consumes less than 1uA current. It has 86 GPIO, 16 PWM channels, SPI, 4 UART, I2C and other features which made it a perfect choice of designing a versatile digital design IC.

Figure 12: Table 8 :

[Md et al. (2013)] , Md , Md. Nasimuzzaman Khaled Hossain , Chowdhury . *International Journal of Emerging Technology and Advanced Engineering* June 2013. 3 (6) p. .

[Zemva et al. (1998)] 'A Rapid Prototyping Environment for teaching Digital Logic Design'. A Zemva , A Trost , B Zajc . *IEEE Transactions* Nov 1998. (4) p. 41.

[Atmega1280 datasheet by Atmel] *Atmega1280 datasheet by Atmel*,

[Digital design principles and practices by John F. Wakerly] *Digital design principles and practices by John F. Wakerly*,

[for simulator and demo videos of simulation] *for simulator and demo videos of simulation*, `http://www.labcenter.com/products/vsm/vsm_overview.cfm` (this IC can become handy)

[Vera et al. ()] 'Integrating Reconfigurable Logic in the First Digital Logic Course'. Guillermo A Vera , Jorge Parra , Craig Kief , Marios Pattichis , Howard Pollard . *9th International Conference on Engineering Education*, july 23-28 2006.

[References Références Referencias © 2013 Global Journals Inc. (US)] *References Références Referencias © 2013 Global Journals Inc. (US)*,

[Mukherjee (2013)] *Remote Control of Industrial Machines Using Mobile Phone*, Poulastya Mukherjee . January 2013. 3. (for project outline diagram)

[Stanisavljevic et al. (2013)] 'SDLDS-System for Digital Logic Design & Simulation'. Z Stanisavljevic , V Pavlovic , B Nikolic , J Djordjevic . *Education, IEEE Transactions on*, May 2013. p. 56.