Design and Implementation of Low Power 16 Bit ALU using clock gating

V. Prasanth
Assoc.prof
Department of Electronics and Communication Engineering
Pragati Engg College, Surampalem, Ap, India

P. Srisuresh
Mtech Student
Department of Electronics and Communication Engineering
Pragati Engg College, Surampalem, Ap, India

Abstract-This paper describes about 16 bit ALU using clock gating. This clock gating generally reduces the consumed amount of power when compared to normal implementation. Clock power reduces 60% of total dynamic power. Theoretically, we can reduce at a maximum of 93.7% of dynamic power. This clock gating internally contains 2 types that are latch based and latch free clock gating in which 88.23% can be reduced using latch based and 70.5% can be reduced using latch free clock gating. The advantage of power reduction is hardware design of the same circuit can be done at 32nm and can be implemented using RTL and TTL level HDL model simulator. After the simulation process the hardware equivalent of the software program using synthesis technique is done using Spartan-3 FPGA.

Keywords —Clock gate, ALU, FPGA, LUT, clock power, register transfer level, dynamic power, leakage power

I. INTRODUCTION

Instruction Decoder unit selects which sub module of ALU is to be executed and which module has to be turned off. This Clock gating Technique reduces at a maximum of 50 to 75% of total chip power.So,either using latch based or latch free clock gating techniques, we can reduce clock power at a rate of 93.75% .The same clock power reduction also used for designing VLSI MOS circuits at 32nm with less consumption with the less frequency applied signals as

$$\text{Power} = C_L \cdot (\text{Voltage}) \cdot (\text{frequency})$$

Power is directly proportional to the square of voltage and the frequency of the clock.

Main circuit to be designed:

To Design a VLSI circuit which consumes less power, Clock is applied only whenever necessary and gates off whenever the power consumption is to be reduced. This is done by Instruction Decoder unit.

II. LITERATURE REVIEW

There are many ways of synthesizing the circuits as circuit level, System level and Register Transfer level. At present the levels are shifted from circuit level to register level using clock gating which switches off the unused sections of the design and reduces the consumed power consumption. Internally it again contains latch based and latch free clock gating design.

Using a Latch free Clock Gating:

At the rising edge of the clock, we use a AND Gate and at the falling edge we use a OR gate as shown in the below figure

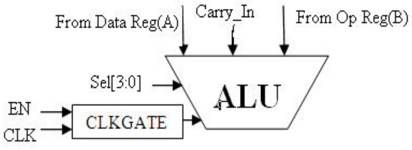

Fig 1 .Latch free clock gated design

Disadvantage in Latch Free clock gating:

As shown in the below figure, if the enable signal goes low before the clock pulse goes to falling edge, the gated clock pulse automatically gets terminated before its actual termination.

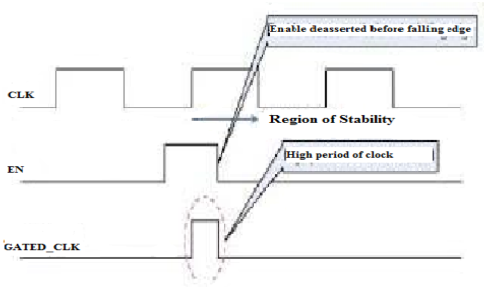

Fig 2.Problem in latch-free clock gated design

Overcomed using Latch based Clock Gated ALU:

In order to overcome the disadvantage in latch free based clock, the enable signal must hold from the active edge to falling edge of the clock, so that gated clock remains active for the complete period

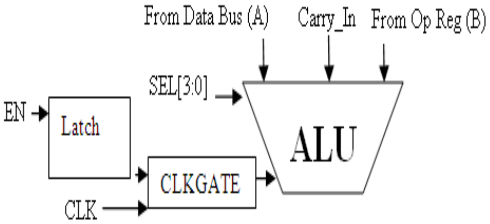

Fig 3. Latch based clock gated ALU design

<table>
<tr><th colspan="4">Functions of Arithmetic and Logic Unit</th></tr>
<tr><th>Unary</th><th>Sel</th><th>Arithmetic &amp; Logic</th><th>Sel</th></tr>
<tr><td>Clear</td><td>0000</td><td>Add</td><td>1000</td></tr>
<tr><td>Hold B</td><td>0001</td><td>Subtract</td><td>1001</td></tr>
<tr><td>Complement B</td><td>0010</td><td>Add Carry</td><td>1010</td></tr>
<tr><td>Hold A</td><td>0011</td><td>Subtract Borrow</td><td>1011</td></tr>
<tr><td>Complement A</td><td>0100</td><td>Logical AND</td><td>1100</td></tr>
<tr><td>Decrement A</td><td>0101</td><td>Logical OR</td><td>1101</td></tr>
<tr><td>Increment A</td><td>0110</td><td>Logical XOR</td><td>1110</td></tr>
<tr><td>Shift Left A</td><td>0111</td><td>Logical XNOR</td><td>1111</td></tr>
<tr><td colspan="2">All Flags are unaffected in execution of Unary except Carry Flag in Shift</td><td colspan="2">All Flag set in every operation from 1000-1111.</td></tr>
</table>

Table I: Function of ALU

For low power consumption ALU, we need a architecture which has a efficient adder for propagation and generation block. The operation of adding will be disabled if we need to perform any logical operation. The same is the method with clock gating also. Here also we switches off the arithmetic operations when logical operation is in use and vice-versa

The functions of ALU are listed above in the table

III. CLOCK GATED ALU

From the Data bus(A),Carry_In and OpReg(B),there are 3 inputs to ALU. The output again returns to the ALU. Opcode [7:4] uses to select [3:0] to select the particular arithmetic or logical operation needed to be performed.

Operations Performed in ALU:

As listed in the above table, the OC[7;4] selects the particular arithmetic or logical operations. The first 8 are unary operations and next 8 are arithmetic and logical operations.

The ALU generates 4 flags-Zero (Z) whose result is zero for any operation performed on 16 bit ALU, Carry (C) generates a carry when any addition on 16 bit or generates a borrow when subtraction is performed on 16 bit data, Sign (S) is generated when the result of any operation is negative, and Parity (P) is generated when the result of operation performed on 16 bit data contains even number of 1's.

Basic IC showing 16 bit ALU

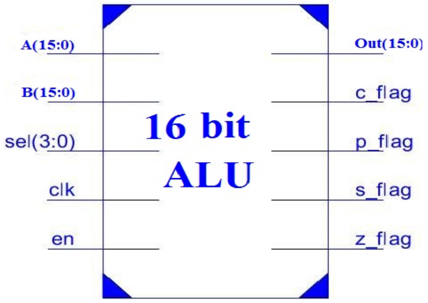

Fig. 5 Arithmetic and logic unit

IV FUNCTIONING OF 16 BIT ALU

A. Clear Function:

It resets the output of ALU to 16’h00.Instead of using a demultiplexed signal, use a clock gate then we can reduce 93.75% power consumption

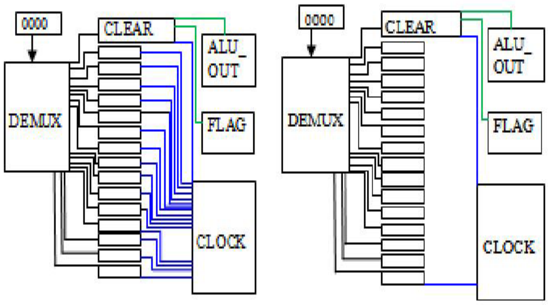

Fig. 6 Clear function

B. Save Operand Register Value in ALU:

This will save the final operand destination value in ALU output section. This can be done by resetting all the other 15 sub modules. Hence 93.75% power can be reduced.

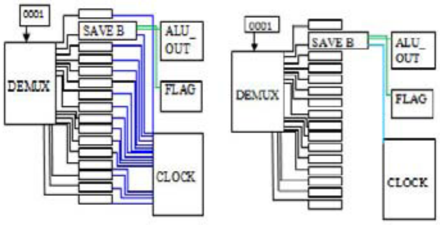

Fig. 7 Save B Operand value

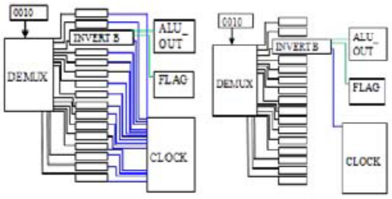

Fig. 8 Invert Operand Register

C. Invert Operand Register Value in ALU:

ALU out= ~B; Pass complemented value of B to ALU output. Hence reduce 93.75% power reduction.

D. Hold Data Bus Value:

ALU out=A; Pass value of A to ALU output.

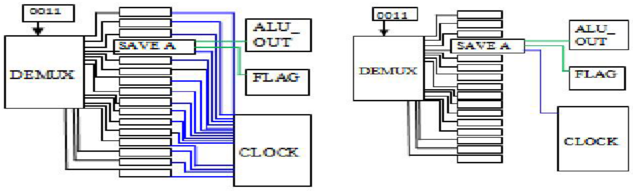

Fig. 9 Hold Data bus value

E. Decrement Data Bus Value:

ALU out=A-1;

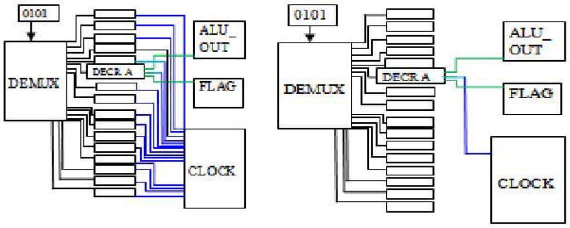

Fig. 10 Decrement Data Bus Value

F. Increment Data Bus Value:

ALU out=A+1;

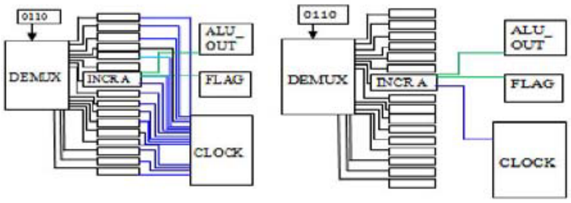

Fig. 11 Increment Data Value

G. Left Shift Data Bus Value:

ALU out=A << 1;

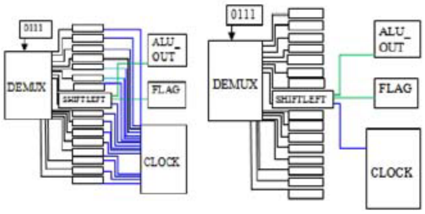

Fig. 12 Shift Left

H. Addition Operation in ALU:

ALU out=A+B;

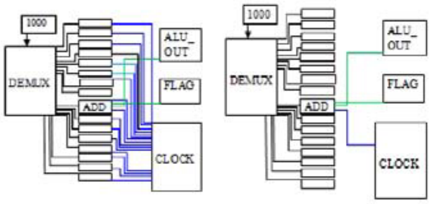

Fig. 13 Addition

I. Subtraction in ALU:

ALU out=A-B;

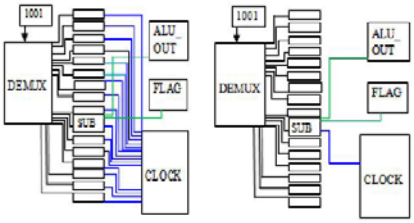

Fig. 14 subtraction

J. Addition with Carry:

ALU out=A+B+Carry_in;

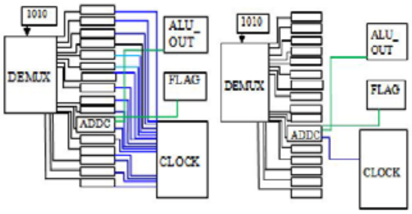

Fig. 15 Addition with carry

K. Subtraction with Carry:

ALU out=A-B-Carry_in;

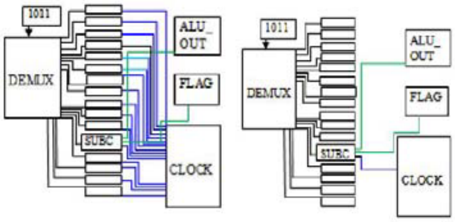

Fig. 16 Subtraction with carry

L. Logical AND Operation:

ALU out=A&B; Calculate Logical A & B and pass that value to ALU out In Clock Gating, we turn off the 15 functional units as shown in Fig.18. Hence reduce 93.75% power reduction

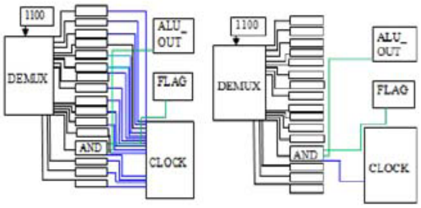

Fig. 17 Logical AND

Similarly remaining operations can be done to reduce power consumption for 93.75%

RTL Technology Schematic:

While synthesising in RTL circuit level,the schematic obtained is as shown.

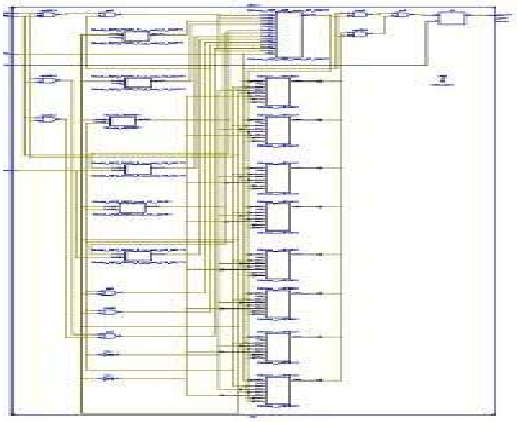

Fig.18 RTL Schematic

V. RESULTS

An low power 16-bit ALU is designed in the platform Xilinx ISE 14.4 and synthesized on 90nm Spartan-3 FPGA.

ALU Power affected by Clock Frequency:

Power is directly proportional to frequency.

frequency	Clock Power	Logic Power	Signal Power	IOs Power
100MHz	2mW	1mW	1mW	0 mW
1000MHz	17 mW	9mW	10 mW	4 mW
10GHz	168 mW	48mW	88 mW	41mW
100GHz	1679mW	153mW	802 mW	410mW
1000GHz	16795mW	1198mW	7983 mW	4099mW

Table 2: Power and clock frequencies

In next phase using clock gating, we turn off rest 15 modules when any module is in execution then theoretical assumption is 93.75% power reduction. Table 3 shows 88.23% clock power reduction using latch based clock gating.

Latch Based Clock Gating	Total Power	Dynamic Power	Clock Power
Without Clock Gate	94mW	41mW	17mW
With Clock Gate	77 mW	25 mW	2mW

Table3: Latch based clock gating

Table 4 shows 70.58% clock power reduction using latch free clock gating.

Latch Free Based Clock Gating	Total Power	Dynamic Power	Clock Power
Without Clock Gate	94mW	41mW	17mW
With Clock Gate	80 mW	28 mW	5mW

Table4: Latch free based clock gating

VI. CONCLUSION

Power consumption has reduced from circuit level to Register level The Register Transfer Level approach is always important because hardware designers generally verify power only at the gate level and any changes to the Register Transfer Level needs many design repetition to reduce power. Our designed ALU has 16 functions. Each function has one dedicated module. When one instruction executes in their respective module, others module that was not used by current executing instruction must gated off by the clock gate. From given formula,

$$\text{Power Reduction \%} = \frac{\text{Number of Unit Gated}}{\text{Total Number of Unit}} \cdot 100$$

hence reduce power (15/16)*100=93.75% power reduction.

VII. FUTURE SCOPE

Using Clock gating we can reduce the dynamic power consumed. We must be able to reduce leakage power.Virtex-6 based on 40 nm technology. Latest FPGA techniques are based on 28 nm technology which contributes certain leakage power. So, there is need to reduce this leakage dynamic power along with dynamic power.

REFERENCES

[1] V. Khorasani, B. V. Vahdat, and M. Mortazavi, “Design and implementation of floating point ALU on a FPGA processor,” IEEE International Conference on Computing, Electronics and Electrical Technologies (ICCEET), pp. 772-776, 2012.

[2] J. P. Oliver, J. Curto, D. Bouvier, M. Ramos, and E. Boemo, “Clock gating and clock enable for FPGA power reduction,” in Proc. 8thSouthern Conference on Programmable Logic (SPL), pp. 1-5, 2012.

[3] J. Shinde and S. S. Salankar, “Clock gating-A power optimizing technique for VLSI circuits,” in Proc. Annual IEEE India Conference

[4] Computer system organization and architecture by William

[5] S. Cisneros, J. J. Panduro, J. Muro, and E. Boemo, “Rapid prototyping of a self-timed ALU with FPGAs,” in *Proc. International Conference on Reconfigurable Computing and FPGAs*, pp. 26-33, 2012.

[6] B. S. Ryu, J. S. Yi, K. Y. Lee, and T. W. Cho, ”A design of low power 16-bit ALU,” in *Proceedings of the IEEE TENCON Conference*, pp.868-871, 1999.