IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 4, April 2015.
www.ijiset.com
ISSN 2348 – 7968

Modified FSM Based 32-Bit Unsigned High Speed Pipelined Multiplier Using Carry Look Ahead Adders In Verilog HDL

Jayalakshmi K R
M-Tech student,
ECE Department,
Mangalam College of Engineering,
Kottayam, India

Ms.Hima Sara Jacob
Assistant Professor,
ECE Department,
Mangalam college of engineering,
Kottayam,India

Abstract— This paper shows a modification to FSM based 32-bit unsigned pipelined mulitipler.It uses carry look ahead adders(CLA's) in place of ripple carry adders(RCA's) in 32-bit FSM based pipelined multiplier.Proposed Multiplier design uses only 1179 slices and 2006 4 input LUT's**.**Synthesis report shows that modified FSM based 32-bit unsigned pipelined multiplier has less delay,less usage of logical resources, than FSM based pipelined multiplier. Simulation was done in Xilinx ISE 13.2(Verilog HDL).

Keywords:
FSM,Pipelined,Carry look ahead adder(CLA),Ripple carry adder(RCA)

1.INTRODUCTION

In many digital system designs Multiplier unit is the main block.The main challenge to the design of a 32-bit multiplier unit is the addition of large number of partial products.There are two main types of hardware implementation method for multiplier unit.Combinational and pipelined implementation. pipelined multiplier Reduces the usage hardware resources by using different pipelined stage.An FSM based 32-bit unsigned high speed pipelined multiplier uses 4 40-bit ripple carry adders(RCA's).The main problem with ripple carry adder is the carry propagation delay.

This paper shows a Modified FSM based 32-bit unsigned high speed pipelined multiplier that uses a carry look ahead adder(CSA). Carry look ahead logic uses the concepts of generating and propagating carries.So the delay has been reduced. The proposed Modified FSM based pipelined multiplier is compared with other multiplier designs.The design method for modified FSM based pipelined multiplier is same as that of existing FSM based pipelined multiplier.we have applied simple shift and add algorithm for the multiplication process. For 32-bit modified FSM based pipelined multiplier there are 32 64-bit partial products. The architecture of FSM based pipelined multiplier and modified FSM based pipelined multipliers almost similar except that the adders are replaced by carry look ahead adders instead of ripple carry adders.

2. LITERATURE REVIEW

FSM based pipelined multiplier has high speed and less usage of hardware resources. But by using Modified FSM based 32-bit unsigned pipelined multiplier further reduction in delay is possible and hence improvement in speed is achieved . It uses carry look ahead adders for the purpose of reducing the carry propagation delay.Other multiplier designs found in the literature has more delay and lower speed than modified FSM based pipelined multiplier.CLA work by creating two signals carry propagator and carry generator.Thus modified FSM based multiplier gives importance to speed of multiplication.

3. EXISTING METHOD

Existing pipelined multiplier design is based on FSM that contain 9 states with less usage of hardware resources.In the architecture three main blocks are present.Partial product generator block,Datapath block and FSM block.It uses ripple carry adders for the addition of partial products.

4.MODIFIED FSM BASED PIPELINED MULTIPLIER ARCHITECTURE

Architecure of modified FSM based pipelined multiplier is divided in to three parts. Partial product generator block,Datapath block and FSM block. In modified FSM based pipelined multiplier we have reduced adder number and added the partial product sequentially to increase maximum operating frequency and reduce hardware resources.

ISSN 2348 – 7968

4.1 PARTIAL PRODUCT GENERATOR BLOCK

This block generate partial product from multiplicand and multiplier.There are 31 different additions process among them 24 additions can be done by 36 bit adder.Each 36 bit adder is a carry look ahead adder.

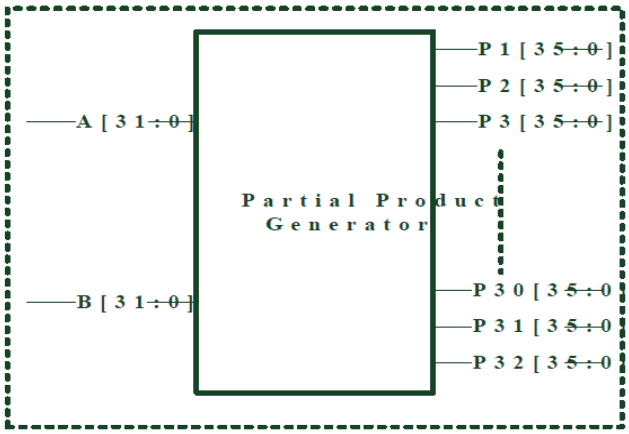

Fig. 1.Modified FSM based pipelined multiplier Partial product generator block[1]

4.2 DATAPATH BLOCK

All the outputs of partial product generator block are assigned as inputs in the Datapath block. In modified FSM we have 32 32-bit partial products. So, we can add the adjacent partial products to reduce to adder width.Here in order to store the carry and to adjust the bit position partial products are modified in each stages. At first stage, we have to change the 32-bit partial product to 34-bit partial product. So there are 16 34-bit additions and that will generate 16 34-bit partial products for second stage. In second stage, we have to change the 34-bit partial product to 36-bit partial product. So there are 8 36-bit additions and that will generate 8 36-bit partial products for third stage. In third stage, we have to again change the 36-bit partial product to 40-bit partial product. So there are 4 40-bit additions and that will generate 4 40-bit partial products for forth stage. In forth stage, we have to modify 40-bit partial product to 48-bit partial product. So there are 2 48-bit additions and that will generate 2 48-bit partial products for fifth stage. In final stage, we have to modify 48-bit partial product to 64-bit partial product. This two 64-bit partial product addition will give 64-bit product.

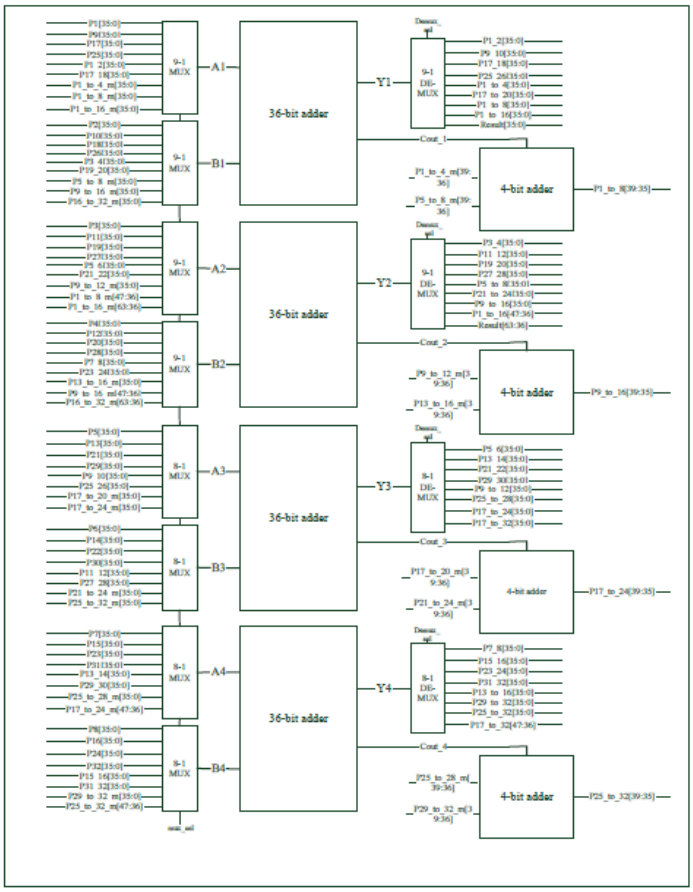

Fig. 2.Modified FSM based pipelined multiplier Data path block[1]

4.3 FSM BLOCK

FSM block generates Mux_sel and Demux_sel input for the Datapath block.FSM contains 9 state and gives Mux_sel [3:0] and Demux_sel [3:0] outputs.

ISSN 2348 – 7968

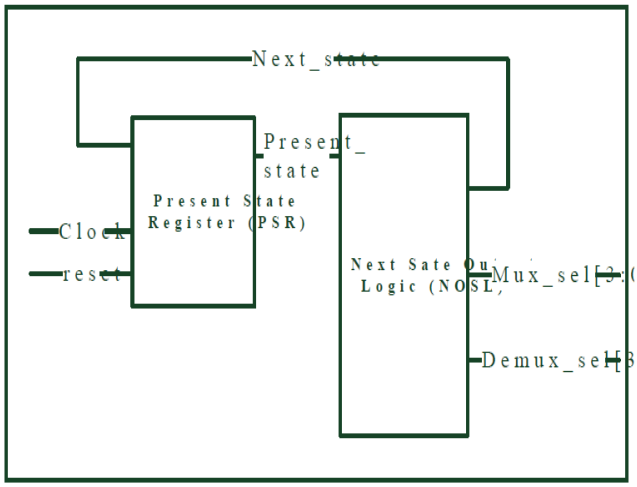

Fig. 3.FSM block of Modified FSM based pipelined multiplier[1]

5.CARRY LOOK AHEAD ADDERS(CLA's) IN MODIFIED FSM BASED PIPELINED MULTIPLIER

Adders are most commonly used in various electronic applications. This is the basic building block of a unit. Different types of adders are available such as ripple carry, carry look ahead, carry select, carry save and many more. Each one having their own benefits and limitations. But the main issue is to design an adder having less delay, low power consumption and reduced chip area. CLA is derived from ripple carry adder. In ripple carry adder data flow in a chain as the bit length go on increasing delay increased to overcome that problem carry look ahead adder was designed. Look ahead carry algorithm speed up the operation to perform addition, because in this algorithm carry for the next stages is calculated in advance based on input signals. Carry look ahead logic uses the concepts of generating and propagating carries. Although in the context of a carry look ahead adder, it is most natural to think of generating and propagating of binary addition, the concepts can be used more generally than this. There will be a carry propagation if OR operation is performed for that either one of the input is one or input carry also be 1.For carry generation there should be AND operation for that both the inputs should be 1. To reduces the carry propagation delay carry look ahead adders(CLA) can be used. They work by creating two signals carry generator and carry propagator.The carry propagator is propagated to the next level whereas the carry generator is used to generate the output carry regardless of the input carry.The integral part of Modified FSM based pipelined multiplier is the partial product addition done by the carry look ahead adders(CLA's .

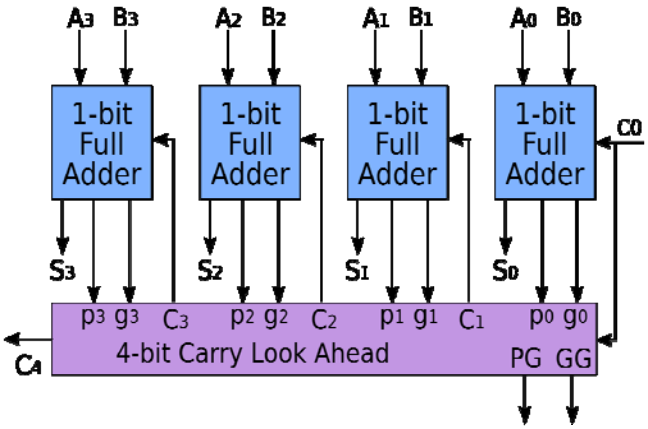

Fig. 4.Carry look ahead adder[2]

Carry-lookahead adder(CLA),improves speed by reducing the amount of time required to determine carry bits. It can be contrasted with the simpler, but usually slower, ripple carry adder for which the carry bit is calculated alongside the sum bit, and each bit must wait until the previous carry has been calculated to begin calculating its own result and carry bits.The carry-lookahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger value bits. The Kogge-Stone adder and Brent-Kung adder are examples of this type of adder.

6.RESULTS

6.1 SYNTHESIS REPORT

Both FSM based pipelined multiplier and Modified FSM based pipelined multipliers are implemented in Xilinx Sparten3 FPGA and the synthesis report shows that our modified FSM based pipelined multiplier has less delay and less usage of hardware resources than existing method.

ISSN 2348 – 7968

```
Timing constraint: Default OFFSET IN BEFORE for Clock 'f3/x1/out_or00001'
  Total number of paths / destination ports: 768 / 128
--------------------------------------------------------------------------------
Offset:              5.889ns (Levels of Logic = 5)
  Source:            in2<1> (PAD)
  Destination:       f3/x2/out_2 (LATCH)
  Destination Clock: f3/x1/out_or00001 falling

  Data Path: in2<1> to f3/x2/out_2
                            Gate     Net
    Cell:in->out      fanout  Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             32  0.821   1.921  in2_1_IBUF (in2_1_IBUF)
     LUT4:I2->O             1  0.551   0.000  f3/x2/Mmux_out_mux0000_935 (f3/x2/Mmux_out_mux0000_935)
     MUXF5:I0->O            1  0.360   0.000  f3/x2/Mmux_out_mux0000_7_f5_34 (f3/x2/Mmux_out_mux0000_7_f535)
     MUXF6:I0->O            1  0.342   1.140  f3/x2/Mmux_out_mux0000_5_f6_34 (f3/x2/Mmux_out_mux0000_5_f635)
     LUT2:I0->O             1  0.551   0.000  f3/x2/sel<3>351 (f3/x2/out_mux0000<9>)
     LD:D                      0.203          f3/x2/out_9
    ----------------------------------------
    Total                      5.889ns (2.828ns logic, 3.061ns route)
                               (48.0% logic, 52.0% route)
```

Fig. 5.Synthesis report showing delay of FSM based pipelined multiplier

```
Device utilization summary:
---------------------------

Selected Device : 3s200tq144-4

 Number of Slices:                     1648  out of   1920    85%
 Number of Slice Flip Flops:             64  out of   3840     1%
 Number of 4 input LUTs:               3074  out of   3840    80%
    Number used as logic:              2946
    Number used as Shift registers:     128
 Number of IOs:                         129
 Number of bonded IOBs:                 129  out of     97   132% (*)
 Number of GCLKs:                         1  out of      8    12%
```

Fig. 6.Device utilization summary of FSM based pipelined multiplier

Table 1. Synthesis report of FSM based 32-bit pipelined multiplier in Xilinx sparten3 FPGA

Clock cycles	Number of slices	Number of 4 input LUT'S	Number of IO'S
9	1648	3074	129

```
--------------------------------------------------------------------------------
Delay:               4.474ns (Levels of Logic = 1)
  Source:            f1/state_FFd1 (FF)
  Destination:       f1/demux_sel_1 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: f1/state_FFd1 to f1/demux_sel_1
                            Gate     Net
    Cell:in->out      fanout  Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FD:C->Q                4  0.720   1.256  f1/state_FFd1 (f1/state_FFd1)
     LUT4:I0->O             5  0.551   0.921  f1/state_FFd1-In1 (f1/state_cmp_eq0007)
     FDS:S                     1.026          f1/demux_sel_1
    ----------------------------------------
    Total                      4.474ns (2.297ns logic, 2.177ns route)
                               (51.3% logic, 48.7% route)

==========================================================================
```

Fig. 7.Synthesis report showing delay of Modified FSM based pipelined multiplier

```
Device utilization summary:
---------------------------

Selected Device : 3s200tq144-4

 Number of Slices:                     1179  out of   1920    61%
 Number of Slice Flip Flops:           1371  out of   3840    35%
 Number of 4 input LUTs:               2006  out of   3840    52%
 Number of IOs:                         129
 Number of bonded IOBs:                 129  out of     97   132% (*)
    IOB Flip Flops:                      64
 Number of GCLKs:                         8  out of      8   100%
```

Fig. 8.Device utilization summary of Modified FSM based pipelined multiplier

Table 2. Synthesis report of Modified FSM based 32-bit pipelined multiplier in Xilinx sparten3 FPGA

Clock cycles	Number of slices	Number of 4 input LUT'S	Number of IO'S
9	1179	2006	129

ISSN 2348 – 7968

6.2 SIMULATION RESULT

Simulation is done in Xilinx ISE 13.2.After assigned inputs, it takes nine clock cycles for the result for both Existing and proposed multiplier designs.

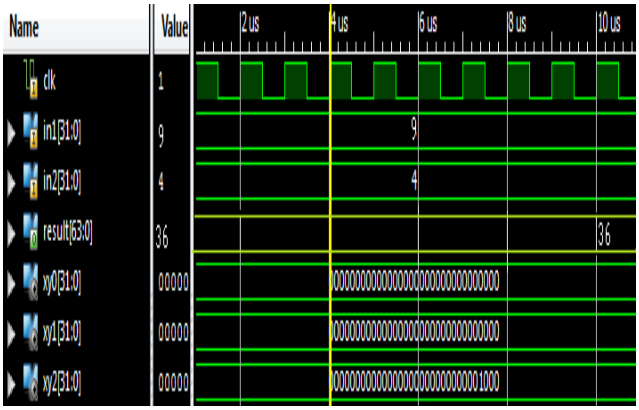

Fig. 9. Timing Diagram of Functional Simulation of FSM based 32-bit pipelined multiplier

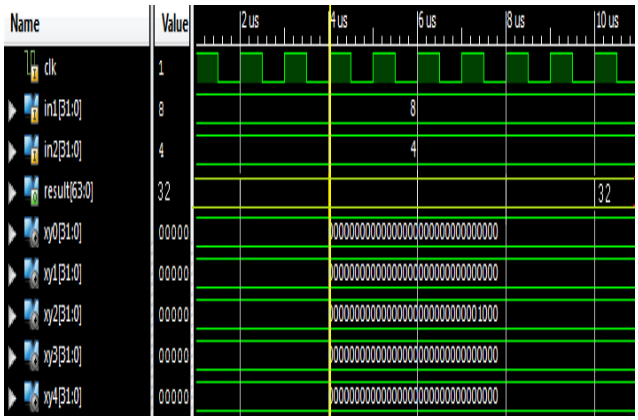

Fig. 10. Timing Diagram of Functional Simulation of Modified FSM based 32-bit pipelined multiplier

6.3 COMPARISON WITH OTHER DESIGNS

Modified FSM based pipelined multiplier is compared with two other multiplier designs along with the existing multiplier design .They are 32-bit signed/unsigned pipelined multiplier and 32-bit Vedic multiplier .Our proposed Modified FSM based Pipelined design gives better output latency and lower critical path delay or lower minimum period than other previous designs.

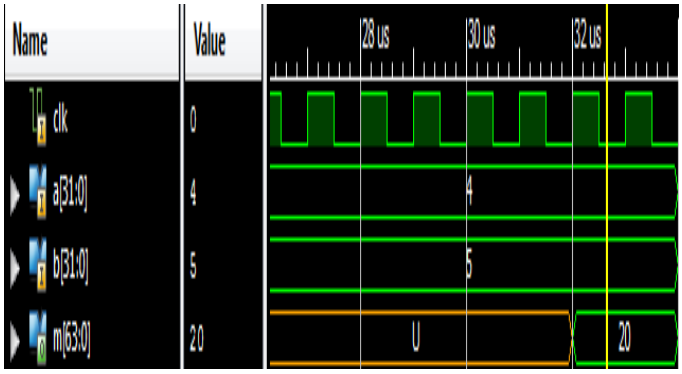

Fig. 11. Timing Diagram of Functional Simulation of 32-bit signed/unsigned pipelined multiplier

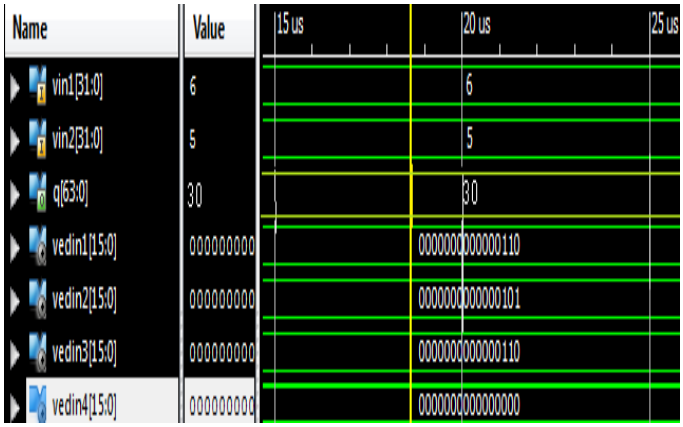

Fig. 12. Timing Diagram of Functional Simulation of 32-bit vedic multiplier

ISSN 2348 – 7968

Table 3. Comparison of critical path delay and output latency with proposed pipelined design and other designs

32-BIT MULTIPLIER DESIGN	OUTPUT LATENCY(Clock cycles)	CRITICAL PATH DELAY(ns)
32-bit signed/unsigned pipelined multiplier	32	6.216
32-bit Vedic multiplier	16	7.165
FSM based pipelined multiplier	9	5.889
MODIFIED FSM BASED PIPELINED MULTIPLIER	**9**	**4.474**

7. CONCLUSION

In this Paper , we designed a Modified FSM based pipelined multiplier which gives better performance than other pipelined design. The design has less output latency to calculate the result and high speed than other design. Synthesis report shows that proposed method has less usage of hardware recourses. Proposed multiplier design was simulated in Xilinx ISE 13.2(Verilog HDL) and done FPGA implementation using sparten3.

8. REFERENCES

[1] Abdullah-Al-Kafi, 1,* Atul Rahman 2, Bushra Mahjabeen 3, Mahmudur Rahman "An efficient design of FSM based 32-bit unsigned high-speed pipelined multiplier using Verilog HDL" 8th International Conference on Electrical and Computer Engineering 20-22 December, 2014, Dhaka, Bangladesh

[2] en.wikipedia.org/wiki/Carry look ahead adders

BIOGRAPHY

Ms.Jayalakshmi K R, received her BTech degree in Electronics and Communication Engineering from Mangalam College of Engineering, Kottayam,India in 2013 and pursuing MTech in VLSI And Embedded system in Mangalam College Of Engineering, Kottayam,India.

Ms.Hima Sara Jacob, Assistant professor at Mangalam College of Engineering,Kottayam,India.She completed her B-Tech in applied electronics and instrumentation from Saint Gits college of engineering,Kottayam,India and Received her M.Tech in VLSI and Embedded system from Saint Gits college of engineering,Kottayam,India.