16x16 Multiplier Design Using Asynchronous Pipeline Based On Constructed Critical Data Path

Authors

Channa.sravya¹, Y.Arpitha², Dr G.Chenchamma³

¹P.G. Scholar, Devineni Venkata Ramana & Dr. Hima Sekhar MIC College of Technology Kanchikacherla, Krishna Dist, PIN: 521180, A.P, India.

²M.E.,(Ph.D), Assistant Professor, Devineni Venkata Ramana & Dr. Hima Sekhar MIC College of Technology Kanchikacherla, Krishna Dist, PIN: 521180, A.P, India.

³Professor, Devineni Venkata Ramana & Dr. Hima Sekhar MIC College of Technology Kanchikacherla, Krishna Dist, PIN: 521180, A.P, India.

ABSTRACT

Asynchronous domino logic pipeline design is a latchless high throughput and low power design. In this design dual-rail domino gates are used to construct the stable critical data path and single-rail domino gates are used in non critical data paths. An 8x8 array style multiplier is used to evaluate this pipeline design. This proposed technique is general and can be used in all domino logic circuit designs. Now this can be implemented for higher order multipliers like 16x16.

INTRODUCTION

All through the latest decade, the main exploration is going on asynchronous technology. Due to increased CMOS technology scaling, VLSI systems complexity is increasing rapidly. Some physical design issues, like global clock tree synthesis and top-level timing optimization, become the genuine issues. Though technology scaling has high integration possibilities, modularity and scalability become tough to be evaluated at the physical level. Asynchronous design is taken as a suitable solution to undertake these problems that relate to the global clock, because local handshake is used in the place of externally supplied global clock [1]–[4].

The attractive properties are listed as follows:

- Power consumption is less;
- Operating speed is high;
- No clock distribution and clock skew problems;
- Better modularity and composability;
- Less emission of electromagnetic noise;

Robustness toward variations in supply voltage, temperature, and fabrication process parameters.

The selection of handshake protocols in asynchronous design influences the circuit parameter like area, speed, power and robustness. The two important protocols used in asynchronous design are four-phase bundled-data protocol and the four-phase dual-rail protocol. The bundled-data design protocol is similar to synchronous design circuits. Local clock pulses are produced by handshake circuits and they use delay
matching to specify accurate signal. Due to the vast use of timing assumptions, efficient circuits can be produced.

Further, the four-phase dual-rail design protocol is implemented in an detailed way that the dual-rail encoding of data is mixed with the handshake signal. Handshake circuits recognize the appearance of genuine data by detecting the encoded handshake signal, which allows correct operation in the existence of data path delays. This characteristic is very helpful in dealing with data path delay variations in advanced VLSI systems, like system-on-chip [3], [4]. Yet, such main feature has encoding and detection overheads. Due to these overheads circuit efficiency is decreased which limit the applications of the four-phase dual-rail protocol design.

This paper presents an innovative design method of asynchronous domino logic pipeline, which targets on increasing the circuit efficiency and making asynchronous domino logic pipeline design more practical for many applications. This innovative design method merges the features of both four-phase dual-rail protocol and the four-phase bundled-data protocol, which gains an area-efficient and ultralow-power asynchronous domino logic pipeline. Asynchronous domino logic pipeline is an pipeline style that can completely avoid explicit storage elements between stages by using the inherent latching function of domino logic gates. This latch less characteristic provides the advantages of decreased critical delays, less silicon area, and less power consumption.

Asynchronous domino logic pipeline has a problem that dual-rail domino logic must be used to construct the domino data path. Due to this domino data path has a dual-rail encoding overhead that takes more silicon area and power consumption. This overhead almost defeats the area and power features provided by the latchless function. Other issue is the overhead of handshake control logic. Common designs of asynchronous domino logic pipeline based on the four-phase dual-rail protocol will depend on domino data path to transfer the data and encoded handshake signal. This domino logic also use completion detectors to detect and gather the handshake signals along the complete data paths [8]–[11].

This design method is very robust for delay variations in data paths though it causes a serious problem in detection. As the width of data paths increases, the detection overhead also increases which blocks its application in the design of a large function block with a data path width. In addition, asynchronous domino logic pipeline based on the four-phase bundled-data protocol avert the detection overhead by considering a single extra bundling signal, to match the worst case block delay, which serves as a completion signal. The main problem in this design method is, it completely fails to use the good properties in the four-phase dual-rail protocol design. Besides, it does not rectify the dual-rail encoding overhead issue in data paths [12].

In this paper, the proposed pipeline decrease both the dual-rail encoding overhead in data paths and the detection overhead in handshake control logic by using constructed critical data path. By redesigning the dual-rail domino gates stable critical data path is constructed. After detecting the stable critical data path, a 1-bit completion detector is sufficient to get the correct handshake signal despite of the data path width. This design greatly decrease the detection overhead and also partially maintains the fine properties in the four-phase dual-rail protocol design.

**LITERATURE SURVEY**


In this paper, the author says about a low power full adder cell with least MOS transistor count that decreases the serious issue of threshold loss. In the middle time, it gradually increases the speed. The simulation has been carried out on Tanner EDA tool on BSIM3v3 90nm and 130nm technologies. The result shows that the technology is not dependent on the circuit.
The idea of the full adder is to develop 3 transistor XOR gate, with the utilization of pass gate property. It is possible to implement the similar adder in the design, especially the adder in 2 bit multiplier. This kind of adder reduces lot of power and delay. But it has disadvantages, like lack of driving capability.


In this paper, author proposed a idea that multiplier can take some errors which allows for quality to be traded off. The inaccurate multipliers have an average power saving of about 40% behind accurate multiplier designs, for an average error of 1.39%-3.32%.

We can use the idea from this paper that creates the method to check error if needed. This will result in some increase of power. In this paper, they have a group of arguments in accuracy and power. So, they can be used in the same way to make arguments about this. Lastly, they implement the multiplier in some DSP applications.


In this paper, explanation is given about variable voltage task scheduling algorithms which minimize energy or minimize peak power. This is a high level voltage scaling description. They determine the connection between the operating voltages for the minimum energy and peak power assignment using the Lagrange multiplier method.

This paper gives some theoretical instructions about the implementation of voltage scaling in terms of algorithms. This will be helpful in dealing tradeoffs between power savings and the complexity of the algorithms.


In this paper various types of full adder designs are proposed. It can be called as a library of different full adder circuits, to explore the trade offs of different adders. It also provides a comparison of performance and power consumption of existing adders. This is useful for adder implementation.

**APCDP TECHNIQUE**

The Figure 1 shows the block diagram of the asynchronous pipeline (APCDP) implemented in 16x16 array style multiplier. This pipeline design is based on a stable critical data path that is build using special dual-rail logic. The critical data path transfers an encoded handshake signal and a data signal. But the noncritical data paths, consisting of single-rail logic, only send the data signal. A NOR gate detects the critical data path and produces a total done signal for every pipeline stage. The outputs of NOR gates are collected at the pre-charge ports of previous stages.

APCDP has the similar protocol like PS0 pipeline logic. The difference between both is that a total done signal is produced by detecting only the critical data path behalf of all the data paths. This design technology has two merits. First, the completion detector is modified to a single NOR gate, and the detection overhead do not grow along with the data path width. Second, the overhead in function block logic is decreased by using single-rail logic in noncritical paths. APCDP has a small overhead in both handshake control logic and function block logic, which greatly increases the throughput and power consumption.
Figure 1: Block diagram of multiplier using APCDP technique.

As shown in Figure 1 indicates critical data path. Each stage in the block diagram is a functional logic block and NOR gate is the detector element. Here four stage pipelining is used which includes fetch, decode, execute and write-back stages. Each stage contains synchronising logic gates (SLG), synchronising logic gates with latches (SLGL), S to D encoding converters. In each pipeline stage, a static NOR gate is used as 1-bit completion detector to produce a total done signal for all the data paths by detecting the constructed critical data path. Driving buffers deliver total done signal to the precharge/evaluation control port of before stages. The single rail domino gates are used in the noncritical data path to save logic overhead.

PS0 LOGIC

PS0 is popular implementation logic of asynchronous domino logic pipeline based on dual-rail protocol [8]. This logic is the basic foundation for many later proposed styles. The proposed pipeline is also based on PS0 pipeline style.

1) Four-Phase Dual-Rail Protocol: PS0 pipeline style is designed based on the four-phase dual-rail protocol. Figure 2 is an example of data transfer depended upon the four-phase dual-rail protocol. The request signal is encoded into the data signal using two wires, \((w_t, w_f)\) in four-phase dual-rail protocol. The data value 1 is encoded as \((1, 0)\), and value 0 is encoded as \((0, 1)\); the spacer is encoded as \((0, 0)\); \((1, 1)\) is not used.

Figure 2: Example of data transfer based on four-phase dual-rail protocol.

2) Structure Of PS0: Figure 3 shows a block diagram of PS0. In PS0 logic, every pipeline stage contains a function block and a completion detector.

Figure 3: Block diagram of PS0.
Every function block is implemented using dual-rail domino logic. The completion detector produces a local handshake signal to control the flow of data through the pipeline. This handshake signal is carried to the precharge/evaluation control port of the previous pipeline stage. A two input NOR gate is used as 1-bit completion detector to generate a bit done signal by observing the outputs of dual-rail domino gate.

To construct a 2-bit completion detector, C-element is used to combine the bit done signals. A full completion detector is formed by gathering all bit done signals from each data path with a tree of C-elements, as indicated in Figure 3.

3) Protocol of Ps0: The PS0 protocol is quite easy and simple. F (N + 1) finishes evaluation when F (N) is precharged. F (N) evaluates when F (N + 1) finishes its reset, or precharge.

In Figure 3, initially data flow through the empty pipeline in which every pipeline stage is in evaluation phase, and the complete cycle of events is as follows.

1) First F1 finishes its evaluation and then the data flows to F2.
2) Data flow to F3 after the evaluation of F2. A precharge signal is sent to F1 after the F2’s completion detector had detected the completion of evaluation.
3) F1 precharges and F3 evaluates. A precharge signal is sent to F2 after the F3’s completion detector had detected the completion of evaluation.
4) F2 precharges. Completion detector of F2 detects the completion of precharge and sends an evaluation signal (enable signal) to F1.

The evaluation signal enables F1 to evaluate new data once again.

SYNCHRONIZING LOGIC GATES:

SLGs are dual-rail domino gates that have no gate-delay data-dependence issue. The principle is that the pull-down network activates exactly one path according to one data pattern, and the remaining all possible data paths are kept constant. Each path has two transistors at the sequential position, and only one path turns ON according to an input data pattern. As a result, the gate delay does not depend on different data patterns. The characteristics of SLGs are:

1) An SLG has a certain number of transistors in pull-down paths at the sequential position.
2) SLG has no gate-delay data-dependence issue and gate delay mainly depend on the input number.
3) An SLG can synchronize the inputs. The SLG cannot start evaluation until all the correct data is arrived.

SYNCHRONIZING LOGIC GATES WITH A LATCH FUNCTION:

Depending on the SLG properties, SLGLs are extended. Synchronizing AND gate with a latch function. An SLGL has an enable port which controls the states of the SLGL. The principle is that SLGLs do not start the evaluation without the enable signal. All traditional dual-rail domino logic can be redesigned to form an SLG or an SLGL. The critical data path in asynchronous pipeline can be easily build using SLGs and SLGLs.

CONSTRUCTION OF THE CRITICAL DATA PATH

A stable critical data path can be build by using the following steps:

- Searching a gate (named as Lin gate) that has the large number of inputs in every pipeline stage;
- Modifying these Lin gates to SLGs;
- Joining these SLGs together to form a stable critical data path.

Algorithm

The design flow is explained using the algorithm.

STEP 1: Consider two inputs A&B of each 16 bit length.
STEP 2: Selecting 2 bits from 8 bits of each input.
STEP 3: Fetching data bits and selecting process.
STEP 4: Decode bits and process to execute.
   Hold step 3 & execute step 2.
STEP 5: Execute the decoded bits.
   Hold step 2 & step 4.
   Execute step 3.
STEP 6: Write back.
   Hold step 3 & step 5.
   Execute step 2 & step 4.
STEP 7: Output for the two input.
These steps are repeated in a loop until all the bits in two inputs are multiplied and the result is obtained finally.

RESULTS
16X16 MULTIPLIER USING APCDP
Example: Multiplication of 32767 X 32767= 1073676289
Timing Diagram

Synthesis Report

Power Analysis

Latency
CONCLUSION

A novel design method of asynchronous domino logic pipeline is introduced. The pipeline is realized based on a constructed critical data path. The design method greatly reduces the overhead of handshake control logic as well as function block logic, which not only increases the pipeline throughput but also decreases the power consumption.

Ripple carry adder circuit is used for MAC architecture. Compared to other circuits, operation speed is more in booth multiplier and it has less hardware count. The basic building units for the MAC are recognized and each of the blocks is analyzed for its performance. Delay and power is calculated for each block. 1-bit MAC unit is designed to reduce the total power consumption depending on block enable technique. Using this block, the N-bit MAC unit is build and the total power consumption is computed for the MAC unit.

Similar power analysis was observed for 8X8 multiplier though number of logic gates is increased in 16x16 multiplier.

REFERENCES