

Enhancing Data Fetching Rates with Parallel Pipelines

Y. Narasimha Rao,
Assistant Professor, GITAM University,
Visakhapatnam

G. Samuel Varaprasada Raju, Ph.D
Professor of Computer Science, Andhra
University, Visakhapatnam

ABSTRACT

In the present techno-world parallel computers are playing vital role in information exchange through various media such as internet and other electronic media. It is very important to consider the data speed along with success rate. The information should reach the destination in time than it become too late. In the present paper, pipeline technologies are discussed to improve data rates. In the present paper two linear pipelines are connected in parallel to improve the fetching speed of the processor. The two pipelines are synchronized and controlled alternatively with common clock pulse.

Keywords: pipeline, data rates, parallelism, parallel pipeline

1. INTRODUCTION

Pipelines plays important role in advanced parallel processor in many ways such as throughput, speed and parallel processing. Before we discuss pipelining it is essential to know about difference synchronizing techniques in pipelining. Synchronous pipeline propagates the data through various stages on single global clock pulse. Which, in contrast asynchronous pipeline do not have any global clock but comparatively have good throughput, speed. Synchronisation can be achieved in Asynchronous pipeline through some effective handshaking signals. In the present work we have proved these characteristics on linear pipeline. A linear pipeline can process successive subtasks with linear precedence. The basic linear pipeline consists of latches and stages. The stages may consist of arithmetic or logic circuits to process required task on the user data or program data. The intermediate latches are used to store the processed data. As the number of stages increases the performance of processor also increases. But there should be a limitation in number of stages because of its complexity in circuit design. The complexity in circuit design may increase propagation delay and further reduces the performance of the processor, which in turn does not satisfy the pipeline requirement.

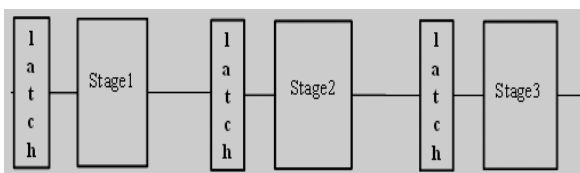


Figure 1: Conventional pipeline

In conventional Pipeline technology the clock signal arrive at the first register take the same time as the data takes time to arrive the first register. But the clock signal may arrive earlier than the data to reach the second stage of the pipeline. So there may be a chance of data overlapping at the second stage of the pipeline. So due to improper synchronization of clock pulses, the traditional pipeline are facing problems such as jitter and skew. In conventional pipeline systems the clock signal is derived as

$$T_{clkconv} \geq D_{max} + D_r + T_s + \Delta_{clk}$$

[2] Where D_{max} is the maximum propagation Delay, D_r is clock-to-output delay of the pipeline register T_s , T_h are the pipeline register setup and hold time and Δ_{clk} is the clock skew at the output register. In the present work a new system is proposed to improve the parallelism. In most of the cases iterations or repeated operations at processing element pays vital role in deciding the speed of the processor. Iterations will consume more time in pipeline than any sequential operations. The next iteration needs to wait till the present iteration finishes its job. Like that in a loop n cycles to execute all iterations. Where n is the maximum number time the task is executing. In many cases repeated operations are performed in sequential order, which in turn consumes more time in loop operations. This will cause degradation in pipeline performance and slows down the processor speed. This delay in looping operations can be avoided by parallel pipelining. With parallel pipelining two operations of iterations can be done in a single clock cycle. If we can execute two similar tasks in a single unit of time, the processor can save large time in long iteration. In the present paper we have shown a simple clock scheme is presented by influencing two concepts in parallel pipelining, one is wave pipelining [3] and hybrid wave pipelining. Wave pipelining is one of the best methods used to minimize the clock skew.

There are already few methods effectively working on clock skew such as wave-pipelining [3] and Mesynchronous pipeline [1] methods. The idea of wave-pipelining [3] was originally introduced by Cotten [6], who named it *maximum rate pipelining* Cotton observed that the rate at which logic can propagate through the circuit depends not on the longest path delay but on the difference between the longest and the shortest path delays. As a result, several computation —waves, i.e., logic signals related to different clock cycles, can propagate through the logic simultaneously. The system clocking must be such that the output data must be clocked into next stage after the data has arrived at the outputs of the present stages. Different clock signal paths can have different delays for a variety of reasons [7][8]. Differences in delays of any active buffers within the clock distribution network may cause un-synchronization of data. Smaller clock periods are achieved in wave pipelining [2] by reducing the maximum propagation delay (D_{max}) by splitting the stages into number of stages. The clock signal is derived in the wave pipelining is

$$T_{clk.w} \geq (D_{max} - D_{min}) + T_h + T_s + 2\Delta_{clk}$$

And further the propagation delay in is reduced and the clock synchronization is controlled by introducing a delay element in the path of clock signal of Mesynchronous pipelining [1]. This delay will be equal to the delay created by the pulse passed from one stage to other stage of the pipeline. The system is clocked such that a pipeline stage is operating on more than one data wave simultaneously [9].

2. EXISTING WORK

A three stage conventional pipeline is constructed with 4 bit registers is shown in figure 2. In traditional pipeline a common clock signal is connected to all registers. The data inputs and outputs are observed in Proteus simulator and simulated results are shown in figure 3. In figure 3 the yellow line represents the

clock signal applied to the individual stages of pipeline. The clock signal is commonly connected to all the stages. The clock signal triggers all the stages simultaneously. This problem may not arise when there are few stages in pipeline. But in the pipeline with more number of stages the clock arrives earlier at last stages when compared with the data arrival time. So simultaneous triggering may causes clock skew in long pipeline. This may lead errors in the output data. The blue line represents the input data and pink wave represents the output data generated through the pipeline stages. As only three stages are used in constructing the pipeline, it is observed that the data propagation delay is almost zero.

The propagation delay can be synchronized with the input data by splitting the stages into multiple stages is called wave pipeline [2]. The stages in traditional pipeline are splitted

to form wave pipeline and is shown in figure 5. As the data propagation delay is small we have applied a common clock pulse.

Simulation results are shown in figure 4. The data transition is occurred at negative edge of the clock pulse. As the stages are divided into small stages the data arrived little earlier at the output of each stage when compared with traditional one. In wave pipeline the next data is fetched before the previous output pulse completely sent out through the pipeline stage. In the figure 4 the output clearly depicting, that at fourth negative edge of the clock pulse (low pulse) the second input data is fetching which is represented in blue colour, while the previous output is under process which is represented in pink colour.

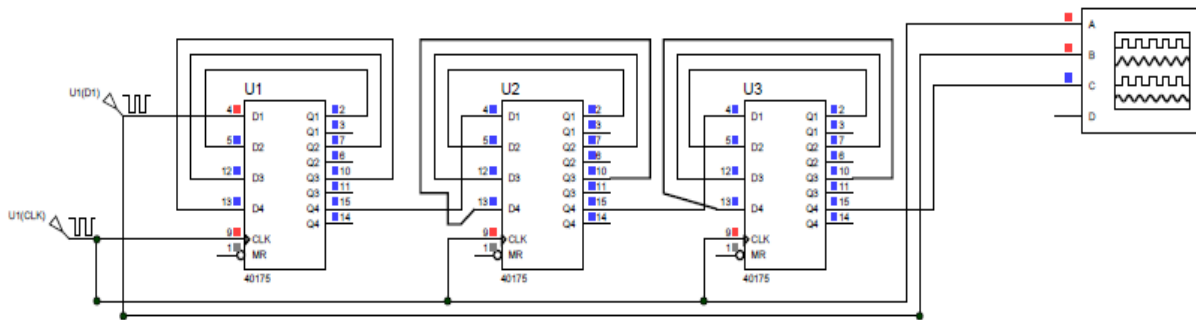


Figure 2: A traditional pipeline with simple logics

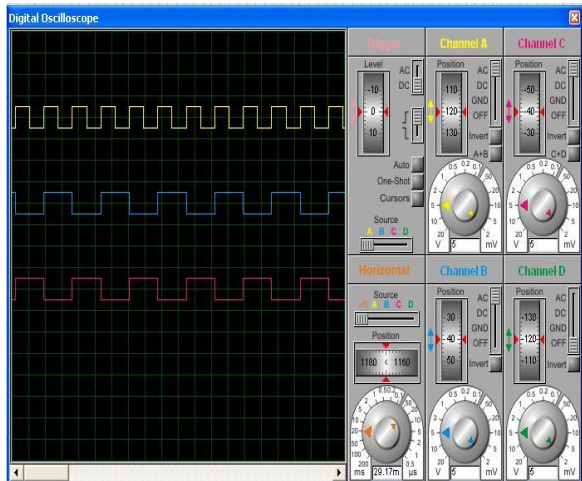


Figure 3: Data wave simulation through traditional pipeline

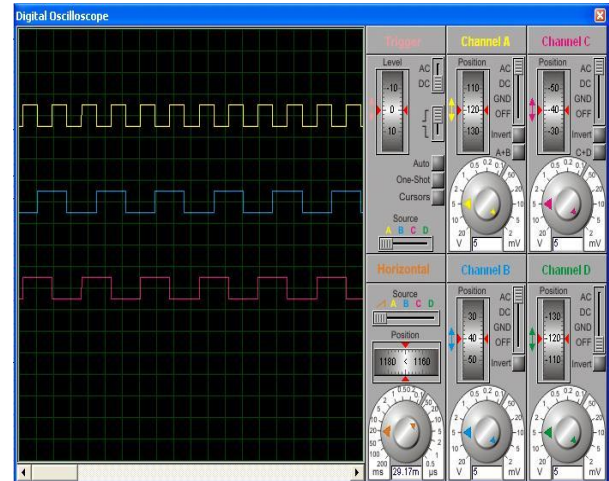


Figure 4: Data wave simulation when pipeline constructed with small stages

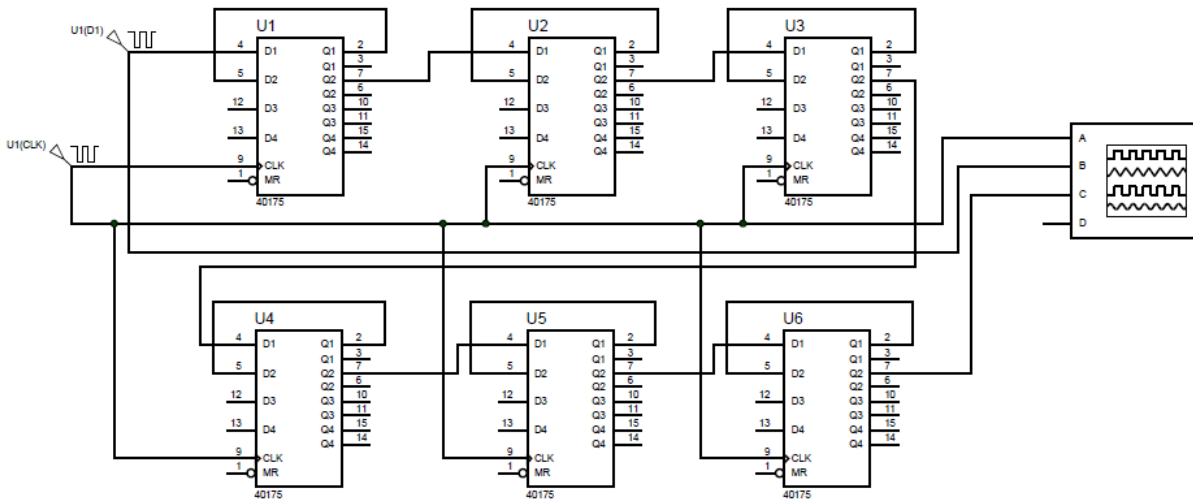


Figure 5: Showing individual stage splitting in pipeline construction

3. PARALLEL PIPELINE METHOD

A two way pipeline is shown in figure 6. It is called 'two way' because the data passes in two ways or two parallel pipelines. Here two pipelines are parallelly connected with a single clock source. The two pipelines are clocked with a common clock signal and they are triggered alternatively. The results are discussed in the next section. In the positive edge of the clock pulse the first pipeline is triggered and at the negative edge of the clock pulse the second pipeline is triggered. This will continue until the clock signal terminated or entered into OFF state.

4. RESULTS

The figure 7 is showing, how the parallel pipelines are processing the same data through two paths of pipelining. The same data will enter into two pipelines almost simultaneously and arrives at the output approximately at the same time. Here yellow colour represents the clock pulse, blue represents the data input, pink colour represents the output data through the first pipeline and green colour represents the output through second pipeline. While the first pipeline processing the present output it is fetching the next data wave and while second pipeline processing the present pulse the first pipeline start producing the alternate pulse. This is done in a small clock period. The two pipeline operations are sharing almost half the clock pulse.

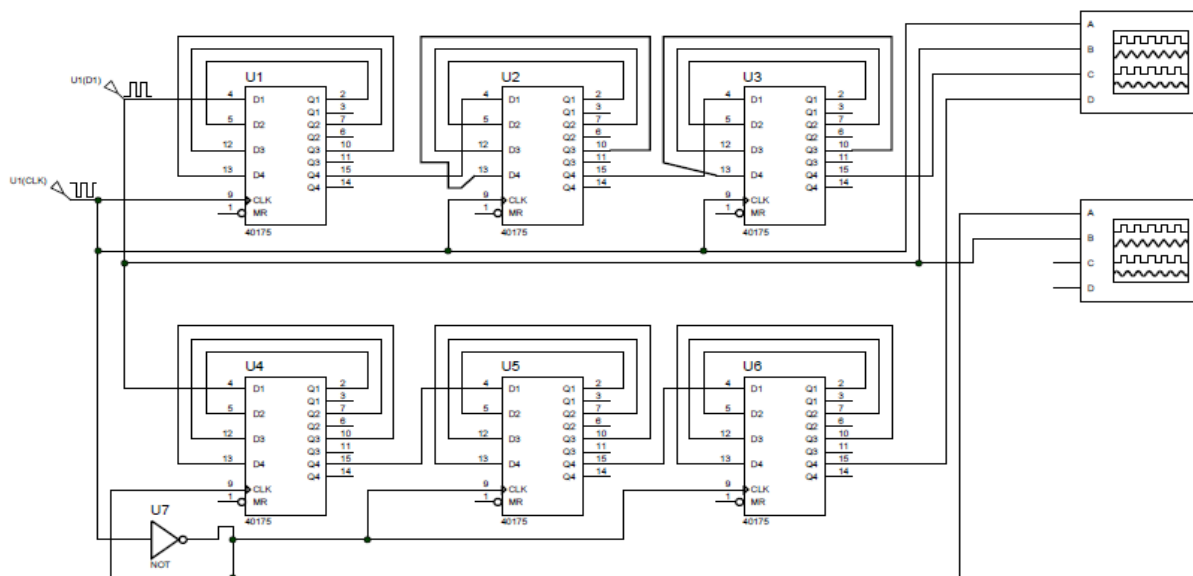


Figure 6: Two way pipeline

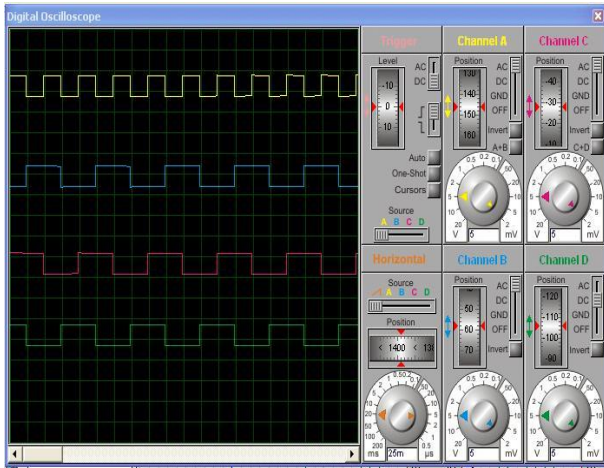


Figure 7: Showing simulation results of Two way pipeline.

5. CONCLUSION

A two way pipeline is designed, simulated and achieved desired output. Two pipelines are parallelly connected and achieved simultaneous data at two outputs. Synchronous clock pulse is applied to each pipeline to trigger pipelines alternatively. The data alternatively gated with pipelines. In this method the data can be sent simultaneously and can process a block twice in a single time. Generally to repeat a block for two times it took two units of time, but here it is done within a unit time. In the present method the data received at second pipeline with a small time delay after the first pipeline output. The pipeline operations are compared with few past methods and shown that in the present method the data fetching rates are improved.

6. REFERENCES

- [1] IEEE and José G. Delgado-Frias, Senior Member, IEEE, A Mesynchronous high performance digital systems, VOL. 53, NO. 5, MAY 2006
- [2] Thomas gay, —Timing constraints for wave pipelined systems| IEEE transactions on Computer aided design of integrated circuits, vol13, no.8, august 1994
- [3] JabulaniNyathi, —A high performance hybrid wave pipelined linear feedback shift register with skew tolerant clocks|, IEEE, 1384- 1387, 2004
- [4] Mohammad Maymandi, —A digital programmable delay element: Design and analysis|, IEEE transaction VLSI systems, Vol.11, no.5, October 2003
- [5] Wayne P. Burleson, —Wave-Pipelining: A Tutorial and Research Survey|, IEEE Transactions on very large scale integration (vlsi) systems, vol. 6, no. 3, september 1998
- [6] L. Cotten, —Maximum rate pipelined systems| in Proc. AFIPS Spring Joint Comput. Conf., 1969.
- [7] Eby G. Friedman, —Clock Distribution Networks in Synchronous Digital Integrated Circuits|, Invited paper, Proceedings of the ieee, vol. 89, no. 5, may 2001 pp665.
- [8] http://www.ece.rochester.edu/users/friedman/papers/Wiley_99_CDN.pdf
- [9] Suryanarayana B.Tatapudi et al., http://www.ece.rochester.edu/users/friedman/papers/Wiley_99_CDN.pdf