RESEARCH ARTICLE OPEN ACCESS

A Survey on Efficient Low Power Asynchronous Pipeline Design Based on the Data Path Logic

D. Nandhini[1], K. Kalirajan[2]
ME[1] VLSI Design, Assistant Professor[2]
Department of Electronics and Communication Engineering
SVS College of Engineering, Coimbatore.
Tamil Nadu - India

ABSTRACT

This paper presents a survey on high-throughput and ultra low-power asynchronous pipeline design method targeting to latch-free and extremely fine-grain design. Since they are asynchronous, these pipelines avoid problems related to high-speed clock distribution, such as clock power, clock skew, and rigidity in handling varied environments. The pipeline communication is structured in such a way that the critical events can be detected and exploited earlier. The survey is mainly done on the data path logic. The data path may be single-rail, dual-rail or combination of the both logic. Asynchronous pipeline based on constructed critical datapath (APCDP) is combination of both the data path. Critical path compose of dual- rail logic and noncritical enables single- rail logic. Based on this critical data path, the handshake circuits are simplified, which offers the pipeline low power consumption as well as high throughput by reducing the overhead problems. This design is going to be implemented by SPICE simulations model.

Keywords:-Asynchronous Pipeline, Dual Rail Logic, Single Rail Logic, Critical Data path.

I. INTRODUCTION

Pipelining is generally used to achieve high performances digital system design. It is classified in to two types as synchronous and asynchronous pipelining. In synchronous system, is a straight forward technique which is used to increase parallelism and hence boost system throughput. Synchronous pipelining design consists of complex functional blocks which are subdivided into smaller blocks; registers are inserted to separate functional blocks. The global clock is applied to all registers. But in asynchronous pipeline design, local clock is applied to all registers [1]-[4]. This pipeline design send data from left to right and an acknowledge control signal is send from right to left that is bidirectional communication, which is implemented by handshaking protocol. Asynchronous pipeline is classified based on the data path logic as static and dynamic logic. Each class uses different approaches for control and data storage. Four important features in providing design flexibility and modularity for asynchronous design are as follow.

First, in synchronous systems, all stages operate at the same fixed rate, and the worst-case stage delay must be less than the clock period. In Asynchronous system, all stages need not have equal delay. Due to dynamically varying delay asynchronous adder have data dependent problem. This should be avoided to improve average system latency and throughput. Second, asynchronous pipelines provide elasticity. Input data items arrive in irregular manner hence the spacing and throughput rate are determined dynamically [5].

Third, asynchronous pipelines provide automatic flow control whereas synchronous pipeline have no flow control. Finally switching activity occurs only when data items are being processed, so dynamic power consumed only on demand. Thus asynchronous pipelining is more efficient than synchronous pipelining. Asynchronous pipelining uses four phase dual rail protocol for handshaking and dual rail or single rail data path logic for valid data passage.

II. DUAL RAIL PIPELINE DESIGNS

A. Williams' PS0 Pipeline Design

Williams' PS0 pipeline [6], which is the starting point of the new pipeline design. Dual- rail is a commonly used method to execute an asynchronous datapath [7][8]. Each pipelining stage consists of a dual –rail datapath, functional block and completion detector. In dual rail, two wires indicate both, the value of the bit and also its validity. The completion detector generate local handshake signal to indicate the presence or absence of data at the outputs of the functional block. The encoding data 00 indicates spacer state, 11 is an unused state. The encoding of 10 and 01 correspond to valid data values 1 and 0 respectively .The protocol of PS0 is simple, single data flow starts with an empty pipeline; the complete cycle of events is as follow:F1 start to evaluate and then the data flow to F2. F2 evaluate and data flow to F3, F2's completion detector detects completion of evaluation and sends a precharge signal to F1.

F1 start precharges and F3 evaluates at the same time [9]. Same cyclic process is repeated for all stage. The analytical cycle time of Williams' PS0 pipeline is given as,

$$T_{CYCLE}=3.T_{EVAL}+2T_{CD}+T_{PREC} \quad (1)$$

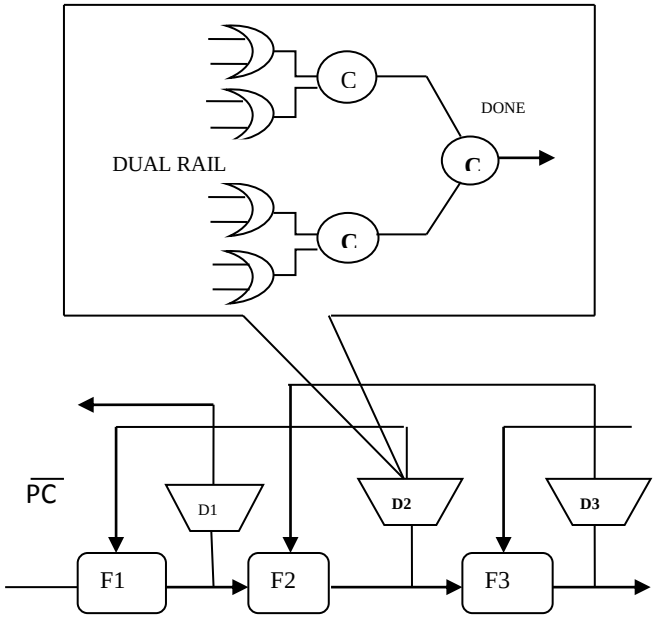

Fig. 1. Williams' PS0 pipeline design

The two main overhead problems of PS0 pipeline are detection overhead in headshake control logic and the dual rail encoding overhead in functional block logic. These problems are solved in LP3/1, LP2/2, and LP2/1 pipeline style.

B. LP3/1 Pipeline Design

In LP3/1 pipeline design style, an early evaluation protocol is used. This protocol receives control information not only from the subsequent stage, but also from its successor stage in pipeline. The key idea of the new protocol is that instead of waiting until N+1 stage has completed precharging, N stage can evaluate as soon as N+1 stage has started precharging [9]. As a result, LP3/1 pipelines have shorter cycles than Williams' PS0 pipelines design but it has longer critical path delay than Williams' PS0 pipelines. The analytical cycle time of LP3/1 pipeline is given as,

$$T_{LP3/1}=3.T_{EVAL}+T_{CD}+T_{NAND} \quad (2)$$

LP3/1 pipeline design will be slower than PS0 pipeline due to greater capacitive loads. Increased loading typically causes logarithmic overheads to the power, area, and latency of the completion detectors. This problem is solved by simply restructuring the LP3/1 Pipeline.

Enhanced Version of LP3/1 Pipeline: Simplifying the stage interfaces

The same protocol is used in Enhanced version of LP3/1 pipeline, but now there is direct communication between adjacent stages. There are two benefits of the simpler stage interface 1) reduced wiring loads, therefore overall wire length and critical path delay is reduced, which can be a significant benefit in future fabrication technologies [10] 2) Greater and easy interfacing with the environment, produces one acknowledgment from the right environment, and one acknowledgment for the left environment. To achieve this two benefit, basic modification is done in pipeline structure first change the NAND gate and second simple redrawing of stage boundaries. Final result is that, rather than using two wires, each stage communicates on only a single wire with its neighbor. So latency is much reduced.

C. LP2/2 Pipeline Design

LP2/2 pipeline design is similar to PS0 pipeline design, but the key difference is completion detectors are now placed before their functional blocks. A modified completion detector generate the "early done" signal. The done signal is passed to the preceding stage when the present stage is about to evaluate (or precharge).Completion detector design is implemented by using an asymmetric C-element [11]. An asymmetric C-element (abbreviated "aC") has three types of inputs: those that are marked "-", those marked "+", and a third type that is unmarked. The output of the aC is set high when all the unmarked inputs and all the "+" ("-") inputs go high (low). The analytical cycle time of LP2/2 pipeline is given as

$$T_{LP2/2}=2.T_{EVAL}+2.T_{CD} \quad (3)$$

D. LP2/1 Pipeline Design

LP2/1 pipeline design combine both the "early evaluation" optimization of LP3/1 pipeline design and the "early done" optimization of LP2/2 pipeline design [9]. Each stage uses information from two succeeding stages (as in LP3/1) LP2/1 pipeline has the shortest analytical cycle time, and also employs early completion detection (as in LP2/2) by this handshake overhead problems are reduced. The analytical cycle time of LP2/1 pipeline is given as.

$$T_{LP2/1}=2.T_{EVAL}+T_{CD}+T_{NAND} \quad (4)$$

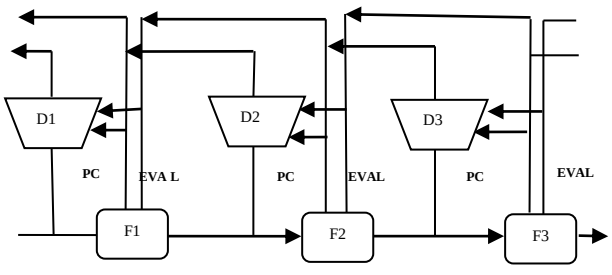

Fig. 2. LP2/1 pipeline design

III. SINGLE RAIL PIPELINE DESIGN

A. Micropipeline Structure

The pipeline structure consists of three components data, control, and latches. The leftmost channel has single rail data and rightmost channel has similar single rail bundled interface. Bundling signal req as input and an ack as output are used as control signal. Delay element is added to each req to match or exceed the worst-case path through the corresponding logic block. The simple chain consists of Muller C-element if inputs are 1, the output is 1, and if both inputs are 0, the output is 0; otherwise, the output maintains its previous value. For storage capture-pass latches is used, which use transition based control signals but provide transparent latch operation. Each latch has two control inputs and outputs. The forward and backward synchronization operations will boost the pipeline delay [5] , and also there are several disadvantages in this design [12], [13].

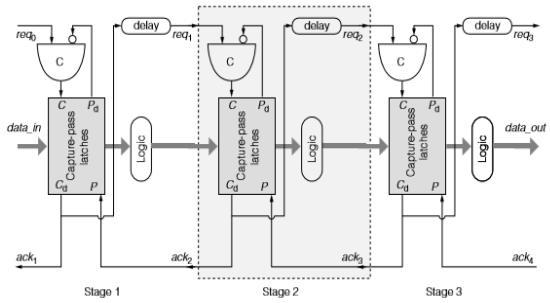

Fig. 3. Static Micropipeline design

B. LP sr 2/2 Pipeline Design

LP sr 2/2 single rail pipeline design is similar to the LP 2/2 dual rail Pipeline design but in single rail single extra "bundling signal" is added sufficiently , to match the worst case block delay, and which serves as a completion signal. Req is the control signal indicates the arrival of new data. If Req signal is high indicates that the previous stage has finished evaluation else Req signal is low indicates that the previous stage has completed precharge .For correct operation timing constrain must be satisfied to met this requirement "matched delay" is inserted [14] which is greater than or equal to the worst case delay through the functional block. An advantage of LP sr 2/2 pipeline design is, data passes through a single rail blocks. A disadvantage is that adequate timing margins, that is added delay must be sufficient to allow the datapath to settle before the request is generated. There are several ways to implement a matched delay. One simply way is to use an inverter chain or chain of transmission gates. Other accurate technique duplicates the worst-case critical path of the logic block, and uses that as a delay line. LP sr 2/2 pipeline is aimed to optimize the cycle time and latency. The cycle time is reduced by *Tap off the early done* signal for the previous stage from *before* to the matched delay, instead of following the matched delay [9]. By *early precharge release*, the functional block can be precharge released before new valid inputs arrive, provided at least that the inputs have precharged, and that new inputs will change only monotonically so the latency can be reduced [15]. The analytical cycle time of LP sr 2/2 Pipeline is given as

$$T_{LP\ SR\ 2/2}=2.T_{EVAL}+2.T_{GC} \quad (5)$$

C. LP sr 2/1 Pipeline Design

LP sr 2/1 pipeline design is a single rail bundled datapath and this design is derived from the LP sr 2/2 pipeline and LP 3/1 pipeline design. Each stage consist of functional block and completion detector same as LP sr 2/2 pipeline design. And each stage receives control signal from its subsequent (PC) and successor (EVAL) stage. LP sr 2/1 Pipeline is a hybrid combination of "early done" and "early evaluation" protocol [9]. Area and power is much reduced by this hybrid combination. It gives better result than all other pipeline design .The analytical cycle time of LP sr 2/1 pipeline is given as

$$T_{LP\ SR\ 2/1}=2.T_{EVAL}+T_{GC}+T_{NANDB} \quad (6)$$

IV. ASYNCHRONOUS PIPELINE BASED ON CONSTRUCTED CRITICAL DATAPATH

APCDP pipeline design is based on a stable critical data path. Noncritical data paths composed of single-rail logic and critical data paths composed of special dual- rail logic. Noncritical path transfer only the data signal, but the critical data path transfers an encoded handshake and data signal. This pipeline design has two merits, first the completion detectors is simplified to a single NOR gate that generates the total done signal for each pipeline stage and the detection overhead is not growing with the data path width. Second, by applying single-rail logic in noncritical data paths the overhead problem of functional block logic is reduced. The noncritical data paths do not have to transfer encoded handshake signal, the completion detector only detects the constructed critical data path. Encoding converter is used as bridge to connect the single-rail gate and dual-rail gate. Constructing stable critical path is very difficult method, because when different inputs are given then critical signal transition varies from one path to other. To solve this problem SLGs and SLGLs is used. Thus the stable critical path is created by finding the gate which has largest input, changing those gates in to SLG logic, then inking those gates to form a stable critical datapath

Synchronizing logic gate SLGs and SLGLs [16],[17] have solved the gate-delay data-dependence and linking problem. APDCP pipeline design has a small overhead in both functional block logic and, handshake control logic, which greatly increase the throughput and

reduces power consumption. The analytical cycle time of APCDP Pipeline is given as

$$T_{APCDP}=2.T_{EVAL}+T_{NOR}+T_{BUFFER} \quad (7)$$

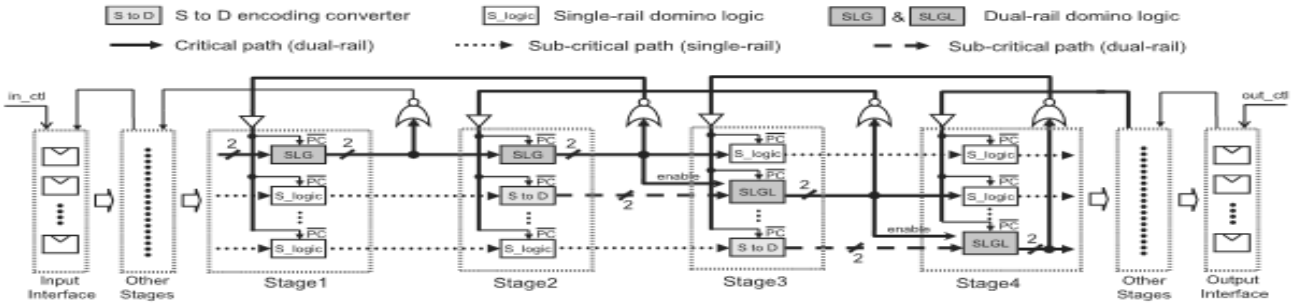

Fig. 4. Asynchronous pipeline based on constructed critical datapath

V. EVALUATION

A. *Experiment Setup*

TABLE I

PERFORMANCES OF DUAL RAIL PIPELINES

Pipeline design	T(eval) (ns)	T(prech) (ns)	T(cd) (ns)	Throughput (Giga item per sec)
PS0	0.21	0.21	0.57	0.51
LP3/1	0.21	0.21	0.60	0.69
LP2/2	0.18	0.21	0.38	0.90
LP2/1	0.18	0.21	0.32	1.04

The results from the above table I indicate that LP2/1 pipeline design delivers the highest throughput of all four designs and its approximately 104% faster than that of Williams' PS0 design .Our other two designs, LP3/1 and LP2/2 also exhibited higher throughputs ,approximately 35% and 76% higher than Williams' PS0 design respectively. In dual -rail pipelining design completion detection consumes 50%–70% of total area and power. The LP2/2 and LP2/1 designs actually had 40% smaller area than PS0 because a pull-up network.

TABLE II

PERFORMANCES OF SINGLE RAIL PIPELINES

Pipeline design	T(eval) (ns)	T(prech) (ns)	T(cd) (ns)	Throughput (Giga/ sec)
LP sr 2/2	0.16	0.18	0.20	1.31
LP sr 2/1	0.16	0.18	0.20	1.55

Static Micropipeline design is a conventional method where the power and area will be very high. And have lot of disadvantage so let's discussion about other two methods. The result from the above table II indicates the time constrain of single rail pipeline. The single-rail data paths were only half as wide, and costly completion detectors were not required. So the single-rail pipeline design were power efficient than the dual- rail pipelines. The single-rail LP 2/2 and LP 2/1 designs consumes 55% lower energy than their dual-rail LP 2/2 and LP 2/1 designs.

TABLE III

EVALUATION RESULTS OF APCDP VERSUS SINGLE RAIL LP2/2 PIPELINE DESIGN

Pipeline design	T(eval) (ns)	T(prech) (ns)	T(cd) (ns)	Throughput (Giga/ sec)
LP sr 2/2	0.16	0.18	0.20	1.31
LP sr 2/1	0.16	0.18	0.20	1.55

The result from the above table III indicates that APCDP reduces transistor count (i.e. area) and FET width. This design has a little larger latency compared with LP sr 2/2 design but this degradation is not serious. Practically APCDP design may have a faster pipeline speed and a lower latency than LP2/2-SR. SLG and SLGL circuits are present only in APCDP design that uses 56 transistors, then to total area will be reduced than other pipeline design. Compared with single- rail logic; APCDP design has efficiently reduces the area, power and also improves the throughput.

B. Result

Fig.5 shows the performances of power consumption for the defined workload. Workload is represented in x-axis and power consumption is represented in y axis. The workload is defined as ratio of the number of active-state cycles to the total number of cycles. The workload is calculated based

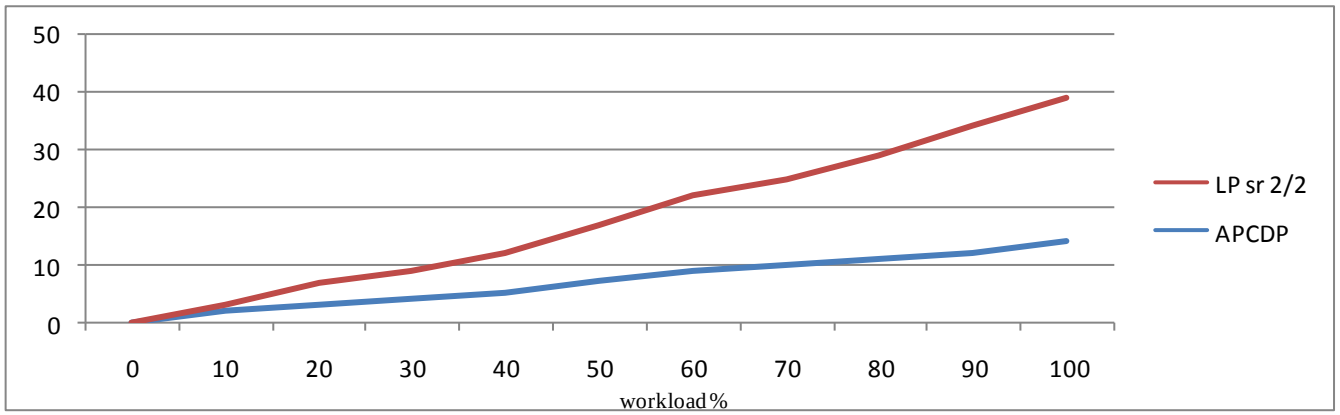

Fig. 5. Performances of power consumption versus workload

on a number of consecutive data insertion cycles (N) following successive empty cycles (M).

$$\text{Workload} = N/(N+M) \qquad (8)$$

As workload increases power consumption also increases, so both power and load are linear to each other. As the result APCDP design is more energy efficient than LP sr 2/2 pipeline design.

VI. FUTURE DISCUSSION

Design automation is an important issue to be considered when applying APCDP to large functional modules, which has to be solved. Static time analysis (STA) is very complex [18] which is to be solved.

VII. CONCLUSION

The survey on the asynchronous pipeline design show that APCDP design method greatly reduces the overhead problem of handshake control logic as well as functional block logic, which not only increases the pipeline throughput but also decreases the power consumption. The evaluation results show that the APCDP design has better performance than a bundled-data (LP2/2-SR) pipeline design.

REFERENCES

[1]. B. H. Calhoun, Y. Cao, X. Li, K. Mai, L. T. Pileggi, and R. A. Rutenbar, "*Digital circuit design challenges and opportunities in the era of nanoscale CMOS,"* Proc. IEEE, vol. 96, no. 2, pp. 343–365, Feb. 2008.

[2]. J. Sparsø and S. Furber, Principles of Asynchronous Circuit Design*: A Systems Perspective.* Boston, MA, USA: Kluwer, 2001.

[3]. M. Krstic, E. Grass, F. K. Gurkaynak, and P. Vivet, "*Globally asynchronous, locally synchronous circuits: Overview and outlook*," IEEE Des. Test Comput., vol. 24, no. 5, pp. 430–441, Sep./Oct. 2007.

[4]. J. Martin and M. Nystrom, "*Asynchronous techniques for system- on-chip design,*" Proc. IEEE, vol. 94, no. 6, pp. 1089–1120, Jun. 2006.

[5]. S. M. Nowick and M. Singh, *"High-performance asynchronous pipelines an overview*," IEEE Des. Test Comput., vol. 28, no. 5, pp. 8–22, Sep./Oct. 2011.

[6]. T. E. Williams, "*Self-timed rings and their application to division*," Ph.D. dissertation, Dept. Electr. Eng. Comput. Sci., Stanford Univ., Stanford, CA, 1991.

[7]. M. B. Josephs, S. M. Nowick, and C. H. K. van Berkel, "*Modeling and design of asynchronous circuits*," Proc. IEEE, vol. 87, no. 2, pp. 234–242, Feb. 1999

[8]. L. Seitz, "*System timing*," in Introduction VLSI Systems, C.A. MeadandL . A. Conway, Eds. Reading, MA:Addison-Wesley,1980, ch. 7

[9]. M. Singh and S. M. Nowick, *"The design of high-performance dynamic asynchronous pipelines: Look ahead style*," IEEE Trans. Very Large Scale Integer. (VLSI) Syst., vol. 15, no. 11, pp. 1256–1269, Nov. 2007.

[10]. International Technology Roadmap for Semiconductors, "Overall roadmap technology characteristics," 2005.

[11]. S.B.FurberandJ.Liu ,"*Dynamic logic in four-phase micropipelines*," in Proc. Int. Symp. Adv. Res. Asynch. Circuits Syst., 1996, pp. 11–16.

[12]. I.E. Sutherland, ‘‘Micropipelines,’’ Comm. ACM, vol. 32, no. 6, 1989, pp. 720-738.

[13]. M. Singh and S.M. Nowick, ‘‘*MOUSETRAP: High-Speed Transition-Signaling Asynchronous Pipelines*,’’ IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 15, no. 6, 2007, pp. 684-698.

[14]. M. G. Peeters, “*Single-rail handshake circuits*,” Ph.D. dissertation, Dept. Math. Comput. Sci., Eindhoven Univ. Technol., Eindhoven, The Netherlands, 1996.

[15]. M. Singh, J. A. Tierno, A. Rylyakov, S. Rylov, and S. M. Nowick, “*An adaptively pipelined mixed synchronous-asynchronous digital FIR filter chip operating at 1.3 gigahertz*,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 7, pp. 1043–1056, Jul. 2010.

[16]. Z. Xia, S. Ishihara, M. Hariyama, and M. Kameyama, “*Synchronising logic gates for wave-pipelining design*,” IEE Electron. Lett., vol. 46, no. 16, pp. 1116–1117, Aug. 2010.

[17]. Z. Xia, S. Ishihara, M. Hariyama, and M. Kameyama, “*Dual-rail/ single-rail hybrid logic design for high-performance asynchronous circuit*,” in Proc. IEEE ISCAS, May 2012, pp. 3017–3020.

[18]. Taubin, J. Cortadella, L. Lavagno, A. Kondratyev, and A. Peeters, *“Design automation of real-life asynchronous devices and systems*,” Found. Trends Electron. Des. Autom., vol. 2, no. 1, pp. 1–133, 2007.