Scheduling Pipelined Circuits

Saikot Das Joy
Dept. of Electronic Engineering
Hochschule Hamm-Lippstadt
Hamm,Germany
saikot-das.joy@stud.hshl.de

***Abstract*—This seminar work is about scheduling pipelined circuits where a circuit is allowed to perform multiple tasks at the same time with maximum overall timing proficiency. in this paper, it is explained what a pipelined circuit is, ways to schedule it, why to use it, the mathematical proof of its efficiency,examples with mathematical computations, area of uses, etc. Different scheduling techniques and comparisons will help us to use the maximum efficiency of hardware components.**

I. Introduction

With the development of science and technology, more and more complicated tasks are easily handled using modern methods and techniques. Hardware components are costly and not easy to build. So, it is always better to make the maximum use of a hardware components. Circuits are the main building blocks of hardware components regarding the functionalities. Pipelining circuit has added a milestone on getting maximum output from a circuit with the fastest time . The main idea behind pipelining is dividing an operation in many stages so that a design can accept new operation even if the previous operation has not been finished completely yet. Pipelining is not using a circuit concurrently, but, It uses the part of a circuit which is unused for another process without interrupting the ongoing execution. Sharing resources like this way has boosted the efficiency of a circuit(e.g CPU) like an unimaginable way. There are different types of pipelined circuits like asynchronous , synchronous which is discussed on this paper with proper explanation and examples. What is pipelining and how it works are also been elaborated. Processes in a pipelined circuits can be scheduled in many ways .This is also discussed on this paper.

II. Pipelining

To get the better understanding of Pipelining, Let's go through an well known example before we move to pipelined circuits.

While washing clothes in washing machine in a dormitory or anywhere else, there is washing machine as well as dryer. Lets consider Figure 1, a washing Machine takes 90 min. for a wash , and a dryer takes 90 min. for a drying.

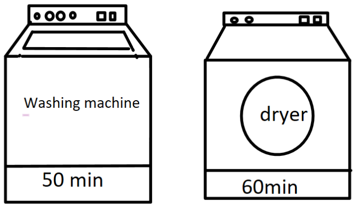


Figure 1. washing machine and dryier

Suppose, there is 10 cloth load,
So, If someone want to wash and dry these all, by normal calculation,
the amount of time necessary =
N*(washing time+drying time) =N*(50+60) =N*110
And this is without pipelining.
But this is not the smartest way. one can easily put 2nd bundle of cloths to washing machine after the first wash while 1st load is drying. Figure 2 explains it very well where this task is done step by step with a pipelined technique:

So,the amount of time necessary in this case would be =N*60 minutes.
It is more like to be (N*60+50).
But we have considered mostly the steady state situation which is our interest in pipelining.
The technique described here is nothing but Pipelining !!

A. Some Definition

Latency: The delay between the establishment of a an input untill the output associated with this input is valid .

for the above mentioned example,for 11 hours(660minutes), if there is no pipelining,
total load that is finished washing and drying = 660/110 = 6 load
while with pipelining , that is = 660/60 = 11 load. it is easily noticible that with pipelining, with the same time, 11 load of cloths are washed and dried!!

Throughput:
throughput is the rate of which input and output are processed.
For this scenario, without pipelining = 1/110 = 0.009 min/sec
with pipelining = 1/60 = 0.016 min/sec

B. pipelined circuits

Let's consider combinational circuit shown in Figure 3. there are several block A,B,C and D , the input of the circuit

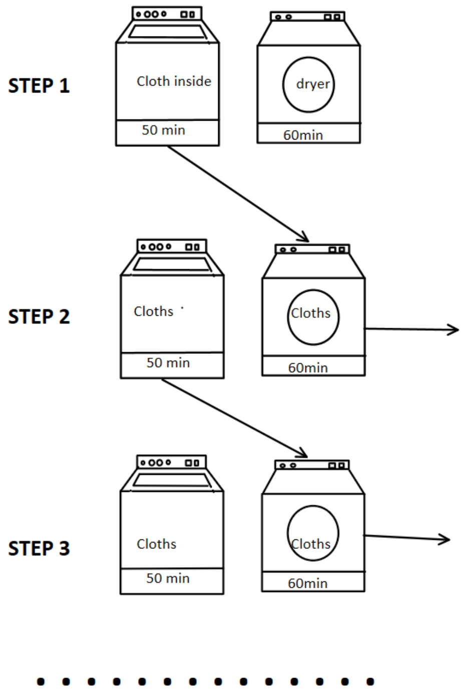


Figure 2. pipelining for using washing machine and dryier multiple time

is X and the output of the circuit is O.

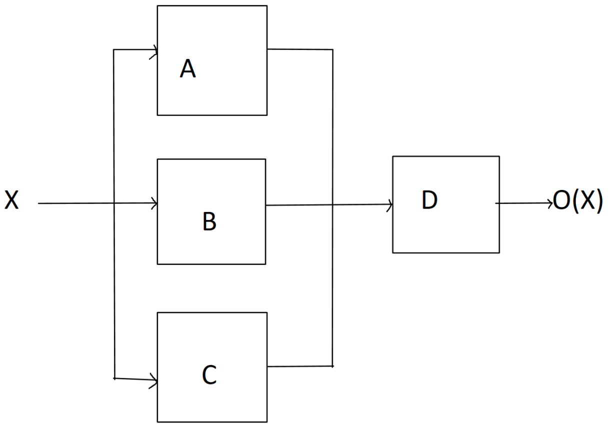


Figure 3. A combinational circuit

Suppose, Latency = L
so throughput = 1/L .
assume that , the processing time of A , B and C is not equal.
Now how this circuit performs a execution ?
in Figure 4, The executional behavious with time is given. It is clearly noticed that, while D performs its execution, A,B and C are idle just kepping the output stable.

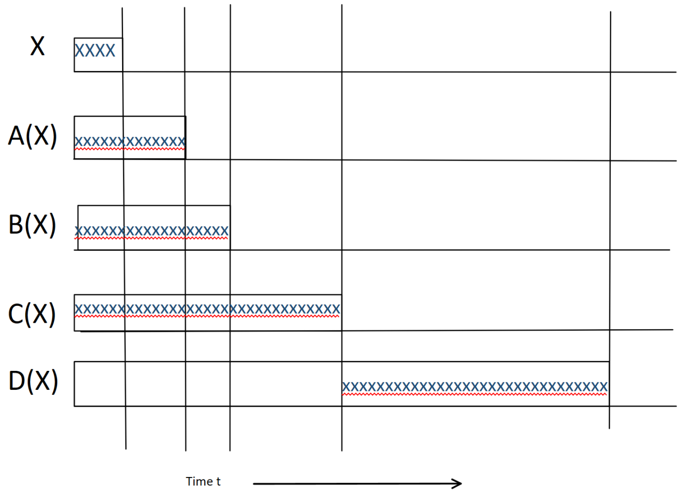


Figure 4. execution of a process in a the combinationa circuit shown in Figure 3

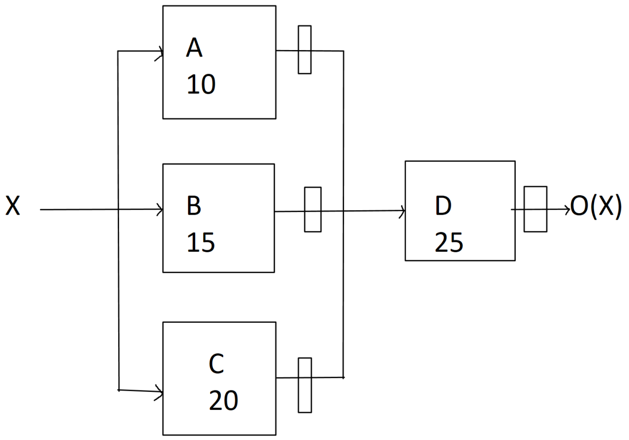


Figure 5. Pipelined circuit

Let's use registers to hold the input values for D stable. and we put some execution time for each elements. Consider Figure 5 where a register is used after each of the blocks shown in Figure 3.

Now, we can consider this combinational circuit as 2 stage pipeline.

A, B and C are working on input

$$X_{i+1}$$

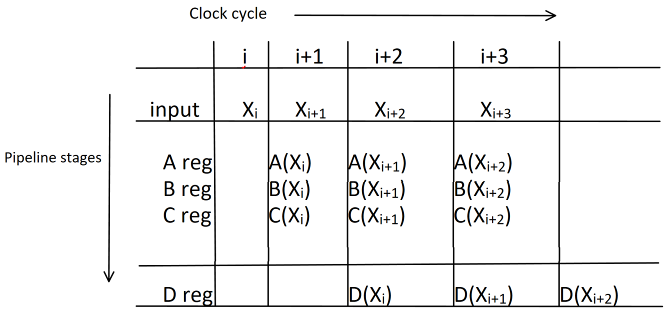
A valid input X during clock cycle j,
O(X) is valid till j+2.

Now, Suppose,
Propagation delays of A, B ,C and D in Figure 5 is 10,15,20 and 25 ns.
An we used ideal zero delay registers.
So, for unpipelined:
Latency= 20+25= 45
Thrughput= 1/45

For two stage pipeline:
Latency= 50
Throughput= 1/25

we see that, though latency is worse in pipelined circuits than the unpipelined circuit, the processing rate which is called throughput is better in pipelined version.

in Figure 6, overall timing behaviour of the circuit shown in Figure 5 is explained .



Clock cycle →

Pipeline stages ↓

	i	i+1	i+2	i+3	
input	X_i	X_{i+1}	X_{i+2}	X_{i+3}	
A reg B reg C reg		$A(X_i)$ $B(X_i)$ $C(X_i)$	$A(X_{i+1})$ $B(X_{i+1})$ $C(X_{i+1})$	$A(X_{i+2})$ $B(X_{i+2})$ $C(X_{i+2})$	
D reg			$D(X_i)$	$D(X_{i+1})$	$D(X_{i+2})$

Figure 6. Pipeline diagram of the circuit shown in Figure 5

for a particular set of inputs data, the result move diagonally through the diagram progressing one pipeline stage for each clock cycle.

C. K-stage pipeline

A K-stage pipeline is an acyclic circuit having exactly K number of register through the path from input to output.
for every pipeline stages, there is an register on it's output.

Latency = K times the period of the clock common to all registers
Throughput = the frequency of the clock

III. Synchronous pipelining

In a synchronous pipelined circuit, all the pipelining stages are controlled by the same clock signal, a global clock.
consider Figure 7, Where pipelining is controlled by global clock, for each stage, input data is processed by combinational logic and output is latched to the input of another stage. The clock duration is large enough which result in filtering hazards or glitch.

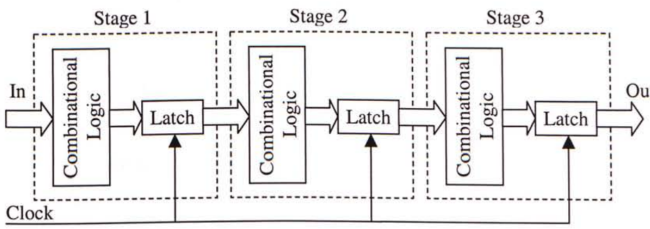


Figure 7. architecture of a pipelined synchronous circuit (from reference [5])

Asynchronous pipelining

In a asynchronous pipelined circuit, all the pipelining stages are controlled by localizes signal instead of global clock signal in synchronous pipelining. Figure 8 shows a classical asynchronous pipelined architecture that makes use of Differential Cascode Voltage-Switch-Logic (DCVSL) as logic cell. here, handshake control signal is used instead of global clock to manage the data flow as well as operation between two pipelined stages. completion of each stages is determined by the data output.

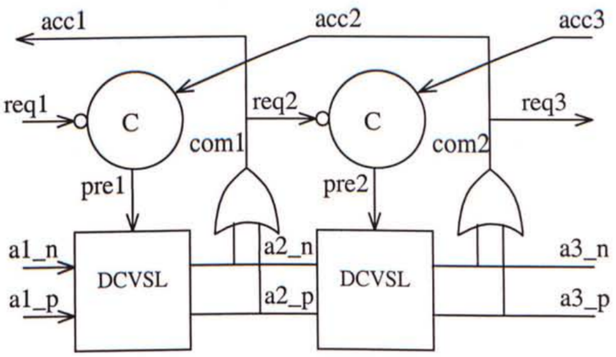


Figure 8. architecture of a pipelined asynchronous circuit (from reference [5])

IV. Scheduling pipeline circuits

generally, a pipelined circuits consists of a sequencial graph model and corresponding data rate.
we consider nonpipelined sequencing graphs whose operations can be bound to pipelined resources. (i.e., pipelined submodels).

In Figure 9, a scheduling example of pipelined circuit or resources is shown.

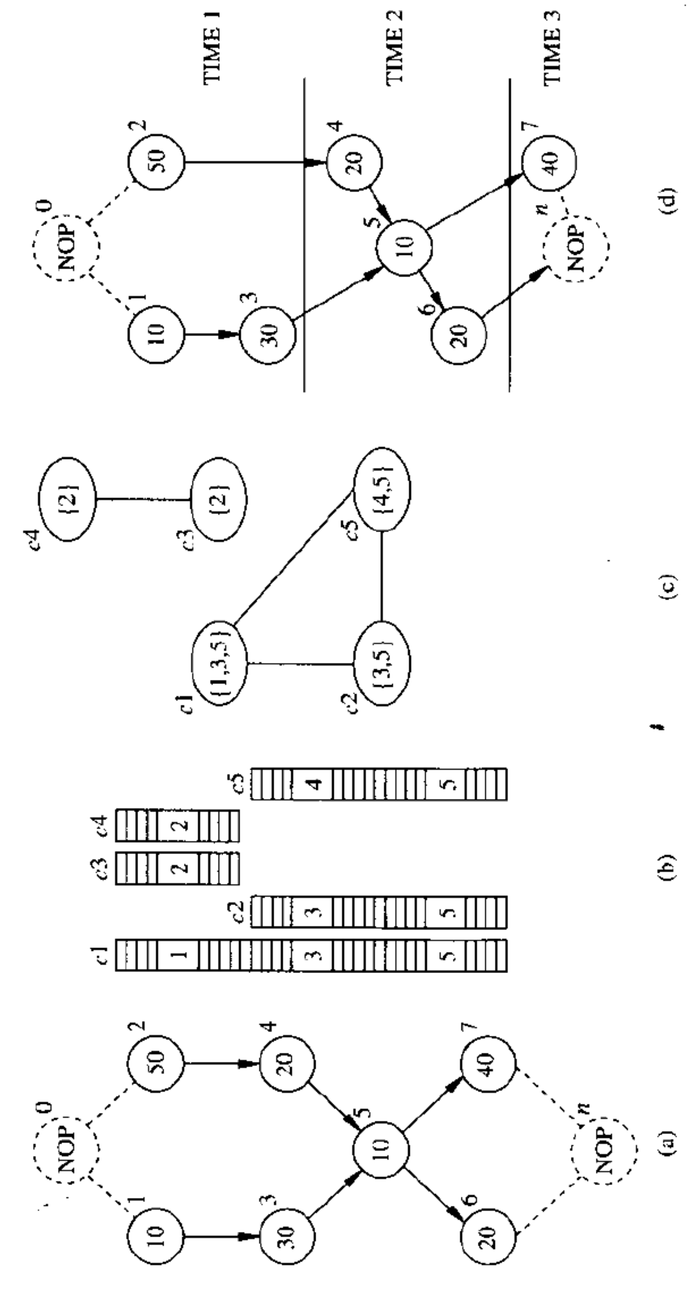


Figure 9. (a) squential graphs (b) cuts (c) cut intersection graph (d) scheduled sequencing graph (from reference [3])

A. Scheduling with pipelined resources

pipelined resources or hardware components or pipelined circuits consume and produces data at time interval less than the execution delay. the time interval which is called data introduction interval are assumed to be constant.
It is possible for pipelined resources to be shared even operations overlap. The fact is, there should be no data dependency, and the operations do not begin at the same time step .
In this case, The list scheduling algorithm can be used to schedule this operation with different time steps and with no data dependency.considering the time introduction interval is 1, the list scheduling algorithm for constraint resources is usuable.
Interested reader can find details of List scheduling on page 207 to 210 of the book on reference [3]. In Figure 10, a scheduled sequencial graph with pipeline resources is shown.One can find that , it is divided into time slices which is called data introduction interval.

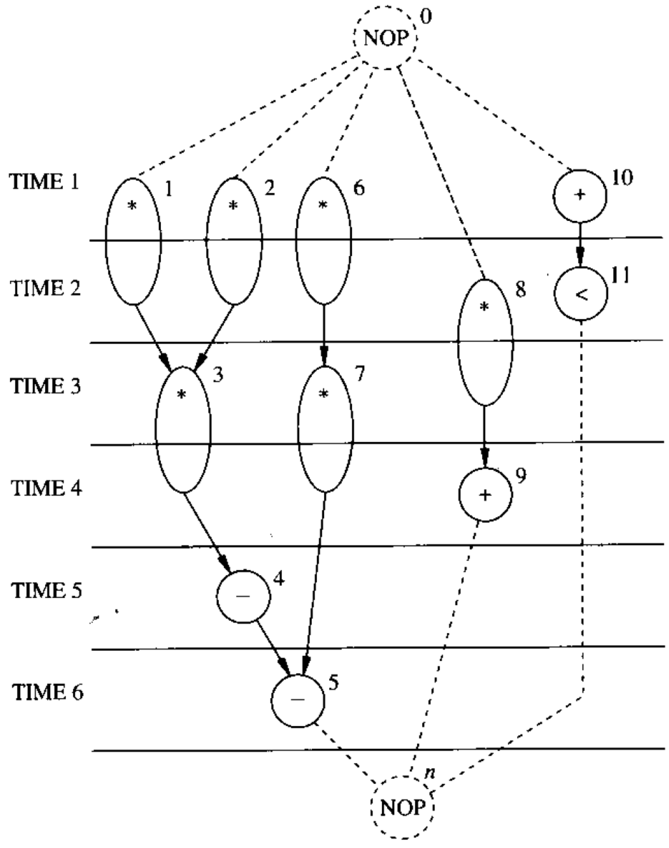


Figure 10. scheduled squential graphs with pipeline resources(image taken from reference [3])

B. Loop folding

Loop folding is an technique which can reduce the execution delay of a loop.we consider the use of pipelining to speed up the execution of loops. This technique is called loop winding and loop folding.
Let's consider a loop,
number of loop iteration = n_l
So, the execution delay of the loop = $n_l.\lambda_l$
where, Latency of the loop = λ_l
pipelining the loop body with local data introduction interval $\delta_l<\lambda_l$, results in:
overall loop execution delay =

$$[n_l + \lambda_l/\delta_l - 1] * \delta_l$$

So, Loop folding reduces execution delay of a scheduled sequency graph that has loop, possibly nested loop. loop folding can be viewed as pipelining the graph entity of the loop body with the goal of reducing the execution delay.

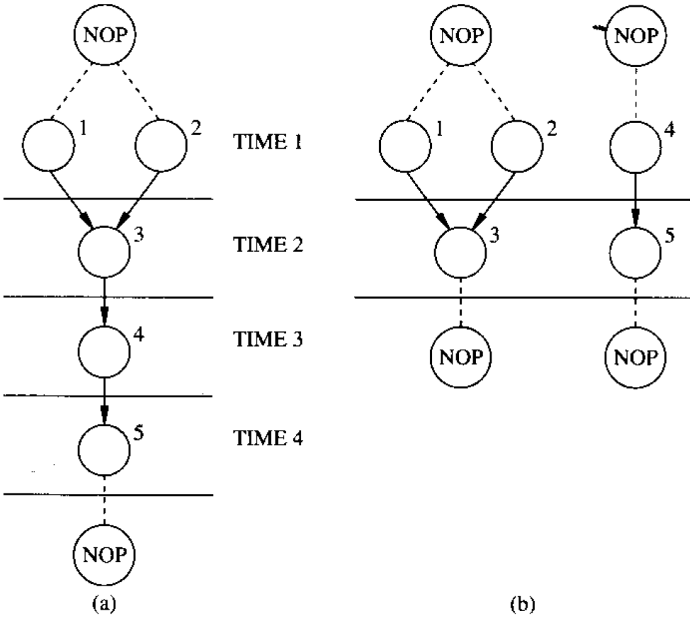


Figure 11. (a) squential graph of a loop body (b)sequencial grapgh after loop folding (image from reference [3])

V. Summery

Increase in the pipeline stages increases the number of execution executed simultaneously. Faster processing unit can be designed using pipeline techniques. By pipelining, electric circuits increases the processing rate of the desired purposed processes. Pipelined CPU works at higher frequency rate . Pipelining increases the overall circuit proficiency.it makes a greater utilization of logic/circuit resources. it is quite exhaustive the act of a pipeline design . it is necessary to divide overall circuits/system into individual stages at an average instances to make sure optimal performance. However, the hard work results in a awesome advantages during the execution.

References

[1] NoLines, Andrew. (1998). Pipelined Asynchronous Circuits.

[2] Yun, Kenneth & Beerel, Peter & Arceo, Julio. (1996). High-Performance Asynchronous Pipeline Circuits.

[3] DE MICHELI, G. (1994). Synthesis and Optimization of Digital Circuits. Berkshire, McGrawhill.

[4] M. -. Marinescu and M. Rinard, "High-level specification and efficient implementation of pipelined circuits," Proceedings of the ASP-DAC 2001. Asia and South Pacific Design Automation Conference 2001 (Cat. No.01EX455), 2001, pp. 655-661, doi: 10.1109/ASPDAC.2001.913384.

[5] TANG,Tin yaw,"An ALU Design using a Novel Asynchronous Pipeline Architecture"(Thesis).The Chinese University of Hong Kong August, 2000

[6] Carbonell, Pablo & Parutto, Pierre & Baudier, Claire & Junot, Christophe & Faulon, Jean-Loup. (2013). Retropath: Automated Pipeline for Embedded Metabolic Circuits. ACS synthetic biology. 3. 10.1021/sb4001273.

[7] R. Kol and R. Ginosar, "A doubly-latched asynchronous pipeline," Proceedings International Conference on Computer Design VLSI in Computers and Processors, 1997, pp. 706-711, doi: 10.1109/ICCD.1997.628942

[8] Keith D. Cooper, Philip J. Schielke, Devika Subramanian, Keith D. Cooper, Philip J. Schielke, & Devika Subramanian. September 1998.An Experimental Evaluation of List Scheduling

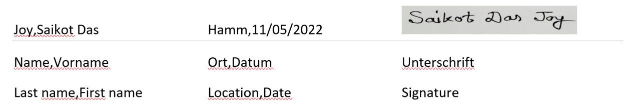
Saikot Das Joy :

Eidesstattliche Erklärung

Hiermit bestätige ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen sowie Hilfsmittel genutzt habe. Alle Ausführungen, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind deutlich kenntlich gemacht. Außerdem versichere ich, dass die vorliegende Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Affidavit

I hereby confirm that I have written this paper independently and have not used any sources or aids other than those indicated. All statements taken from other sources in wording or sense are clearly marked. Furthermore, I assure that this paper has not been part of a course or examination in the same or a similar version.



Joy,Saikot Das	Hamm,11/05/2022	Saikot Das Joy
Name,Vorname	Ort,Datum	Unterschrift
Last name,First name	Location,Date	Signature