Lecture 28: April 27, 2021

Computer Architecture and Organization-I

Biplab K Sikdar

0.2 Arithmetic Pipeline

Floating point addition/multiplication/division can be divided into number of phases.

Example pipeline for fixed point multiplication:

Multiplication logic used here is partial product logic.

Figure 15 is the 6-bit multiplier. It produces 6 (12-bit) partial products.

Each partial product is shifted left. The i^{th} partial product is shifted left i times.

Carry save adders, shift units and carry look ahead adder are used in a pipeline.

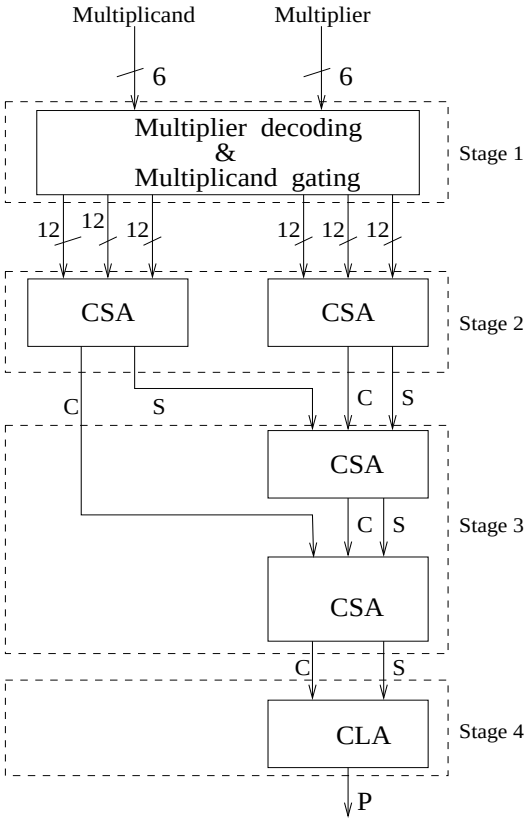

Figure 15: Arithmetic pipelining

Pipeline hardware for multiplication contains 4 stages.

Stage 1: contains logic for generating 6 partial products with appropriate shift.

Stage2 receives 6 partial products and generates 4.

Stage3 converts this into 2 partial products.

Stage 4: has a CLA, which finally generate product.

Here, the total gate delays in each stage are not same.

A design with almost same delay in pipeline stages is desirable.

A better design is shown in Figure 16.

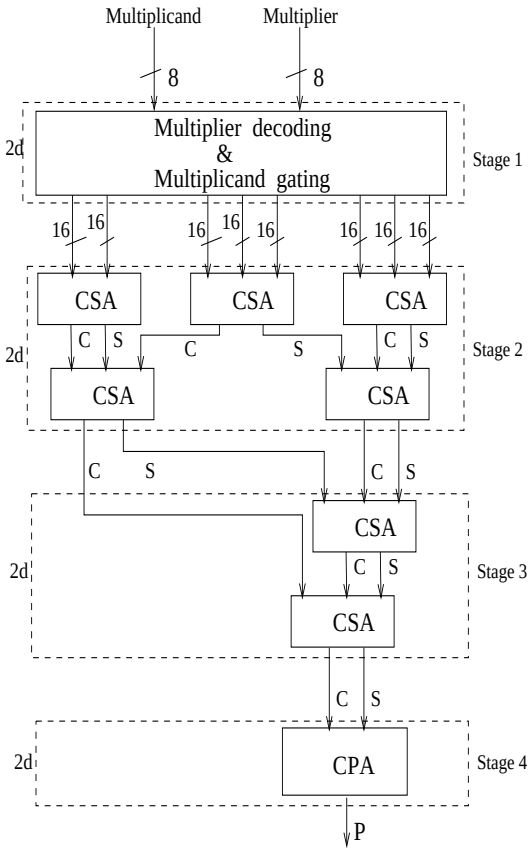

Figure 16: Arithmetic pipelining - 8-bit multiplication

0.3 Processor Pipeline

In processor pipelining, each stage of pipeline is a processing element (PE).

It performs different operations on same data stream by a cascade of processors.

This class of architecture is the multiple instruction single data (MISD) machine.

0.4 Linear Pipeline

Processing stages are cascaded linearly (Figure 17).

No feedback or feedforward path among pipeline stages.

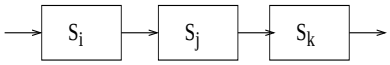

Figure 17: Linear pipeline

Asynchronous and synchronous models:

Delay of a pipeline stage S_i may not be same as that of stage S_{i+1}.

If delay of S_i is less compared to S_{i+1}, then S_i must hold its generated output so that S_{i+1} gets ready to accept this as input.

Data flow from stage S_i to S_{i+1} can be synchronized with a clock - synchronous pipeline (Figure 18).

Latches are introduced to handle delay mismatch.

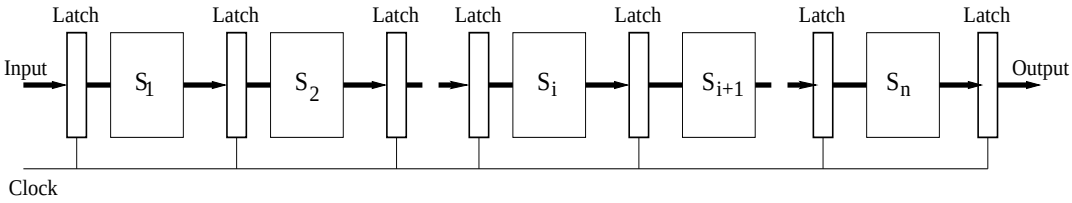

Figure 18: Synchronous pipeline

Utilization of pipeline stages (S_1, S_2, $\cdots$, S_i, S_{i+1}, $\cdots$, S_n) in different clock cycles can be specified by reservation table of pipeline function.

In asynchronous pipeline, data transfer from stage S_i to S_{i+1} can only be realized when stage S_{i+1} is ready to accept data from S_i (Figure 19).

This follows a handshaking protocol.

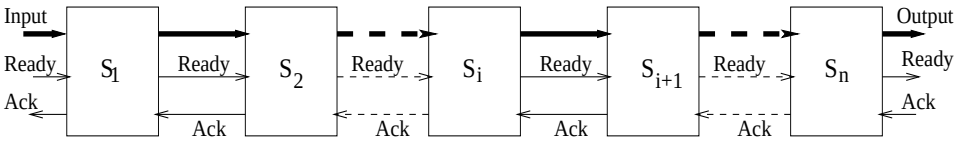

Figure 19: Asynchronous pipeline

0.5 Non-linear Pipeline

A pipeline architecture can reconfigure different functions at different times.

Pipeline architecture of Figure 20 (Figure 21) can realize a function say, F1 that utilizes S_1, S_2, and S_3 in 3 consecutive clock/pipeline cycles respectively.

Function F2 may follow S_1, S_2, S_3, S_1, and S_2. It requires 5 clock cycles.

Such pipeline structure (allows feedback/feedforward), is a *nonlinear pipeline*.

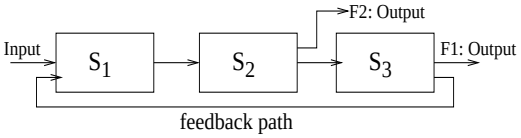

Figure 20: Feedback

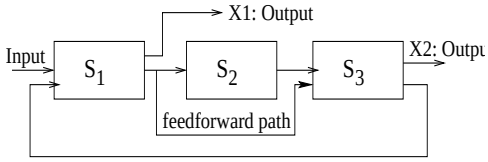

Figure 21: Feedforward

Features of nonlinear pipeline:

a. These are multifunctional pipelines.

b. There can be feedback from a later stage j to an earlier stage i of the pipeline (in Figure 20, from stage S_3 to S_1).

c. There can be feedforward links from pipeline stage i to stage j (in Figure 21, from stage S_1 to S_3).

d. Pipeline logic controls links/paths that are to be activated at a given cycle.