

Sequential (Seq) Circuit: Large Design

- Large Seq Circuit is made up of a data path and a control unit.
- Data path consists of seq and combinational circuit such as registers, counters, mux, decoders and ALU'S. (Arithmetic Logic Unit)
- High Clock frequency (freq) implies data path has a short maximum propagation delay.
- Max clock freq results in higher core junction temperature.
- Designer's goal is to stay within/under max Tj (Tj= temperature inside Computer Brain) to avoid signal integrity issues.

Register transfer notation (RTN) is used to describe an operation of a data path.

- **Formally describes a data path operation**
- **May use an arbitrary or an HDL syntax**
- **Examples:**
 - $CNTR \leftarrow CNTR + 1$ //incrementing counter
 - $CNTR \leq CNTR + 1;$ //Verilog HDL
 - $R \leftarrow R[7] // R[7:1]$ //Arithmetic right shift
 - $R \leq R >>> 1;$ //arithmetic right shift (Verilog)
 - $R \leq \{R[7], R[7:1]\};$ //arithmetic right shift (Verilog)
 - $M[x] \leftarrow R;$ //memory transfer (write)
 - $R \leftarrow M[x];$ //memory transfer (read)
 - Etc.

- Architecture of data path can be classified as single cycle, multiple cycles or pipelined.
- A single-cycled data path requires more hardware but a simpler control unit.
- A multicycle data path requires less hardware but generates results in several clock cycles.
- A pipelined data path also requires more hardware but can operate on multiple inputs concurrently.
- The single Data path contains two adders (+) modules and one adder/sub tractor (+/-) module.
- The single mode controls the functions of the adder/sub tractor modules.
- Time period is proportional to the propagation delay of the longest signal path that starts from the inputs of the first adder and ends at the input of the register.
- In general, a single cycle data path implements several simple and complex operations, its minimum clock would be proportional to the time required to complete most complex operations.

Sequential Circuit: Single cycle Datapath (Timing)

T_{cq}: Clock to queue, the time the flip-flop takes to change its output after a (rising) edge of clock or clock edge.

T_{st} (T setup time): is defined as the minimum amount of time before the clock's active edge that the data must be stable for it to latch correctly.

T_{cs}: clock to skew (sometimes called timing skew) is a phenomenon in synchronous digital circuit systems (such as a computer systems) in which the same sourced clock signal arrives at different components at different times

T_{hold} or T_h: Hold time is defined as the minimum amount of time after the clock's active edge during which data must be stable.

Sequential circuit: single cycle data path

- Data path contains two adder modules and one adder/subtractor module
- The single mode controls the functions of the adder/subtractor modules
- Time period is proportional to the propagation delay of the longest signal path that starts from the inputs of the first adder and ends at the input of the register.
- In general, if single cycle data path implements several simple and complex operations, its minimum clock would be proportional to the time required to complete most complex operation.

Sequential circuit: Single cycle data path architecture

-Data path that computes either the quantity

- $A + B + C + D$ or $A + B + C - D$

-Equation that estimates the minimum clock period (τ) required to run the data path

- Add stands for Adder; Sub stands for subtractor
- Δ is delta time delay from input to output

$$\tau_s \geq 2\Delta_{\text{add}} + \Delta_{\text{add/sub}} + T_{\text{st}} + T_{\text{cq}} + T_{\text{cs}}$$

$$\tau_s = \tau_{\text{single-cycle}}$$

Sequential circuit: Multicycle data path architecture

-Data path that computes either the quantity

- $A + B + C + D$ or $A + B + C - D$

-Equation that estimates the minimum clock period (τ) required to run the data path

- Add stands for Adder; Sub stands for subtractor; Mux stands for Multiplexor
- Δ is delta time delay from input to output

$$\tau_m \geq \Delta_{\text{mux1}} + \Delta_{\text{add/sub}} + \Delta_{\text{mux2}} + T_{\text{st}} + T_{\text{cq}} + T_{\text{cs}}$$

$$\tau_m = \tau_{\text{-multicycle}}$$

-A multicycle data path requires that a computation be divided and computed in steps.

-A multi cycle algorithm to implement $R \leftarrow A + B + C + D$ or $A + B + C - D$; (5 possible simple operations)

Cycle 1: $R \leftarrow A$

Cycle 2: $R \leftarrow R + B$

Cycle 3: $R \leftarrow R + C$

Cycle 4: If mode == 0, then $R \leftarrow R + D$; otherwise $R \leftarrow R - D$

Seq circuit: Pipelined Data path architecture

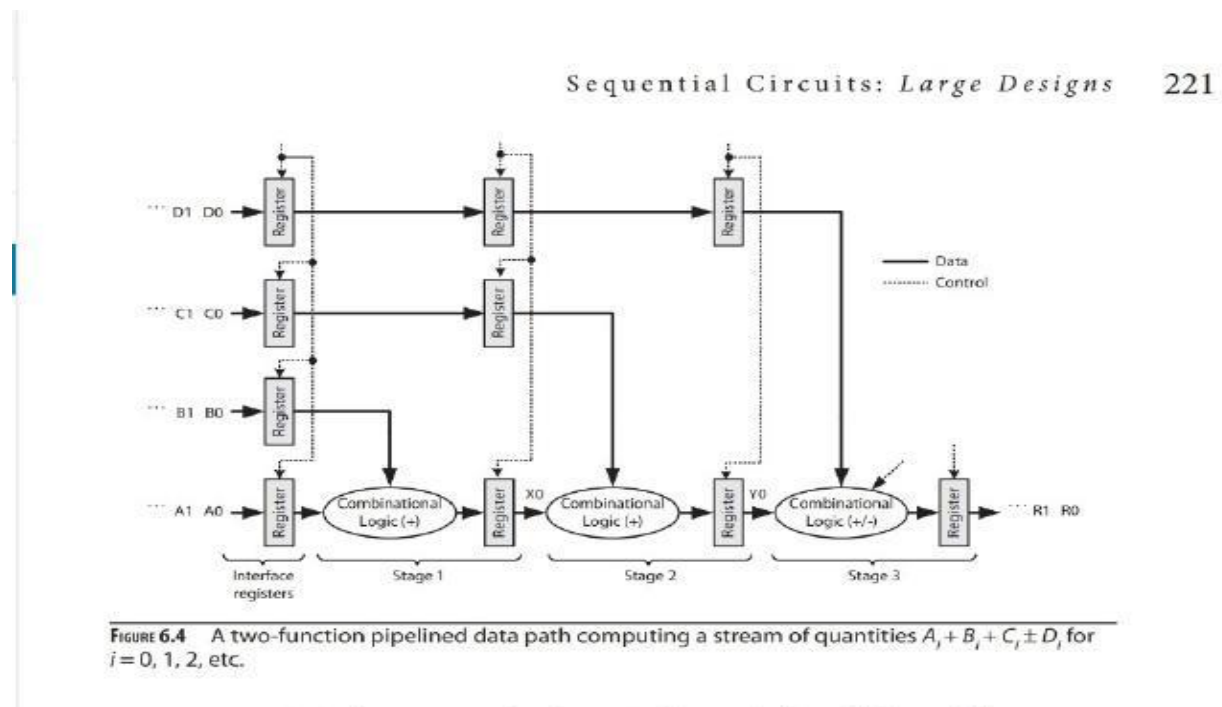
-Data path that computes stream of quantities $A_i + b_i + C_i +/- D_i$

- $A_i + b_i + C_i +/- D_i$
- τ = represents minimum clock period for pipeline data path architecture
- Equation that estimates the minimum clock period (τ) required to run the data path
 - Add stands for Adder; Sub stands for subtractor; Mux stands for Multiplexor
 - Δ is delta time delay from input to output

$$\tau_p \geq \Delta_{\text{add/sub}} + T_{\text{st}} + T_{\text{cq}} + T_{\text{cs}}$$

$$\tau_p = \tau_{\text{pipeline}}$$

- Computing stream of quantities $A_i + b_i + C_i +/- D_i$.



- Seq circuit: Pipelined Data path architecture (Cont)

Refer to Figure 6.5 -Horizontal Pipeline chart

A pipeline uses more hardware, similar to a single-cycle data path, but operates with a higher-frequency clock, similar to a multicycle data path. Furthermore, it can process a stream of data a lot faster than the other two data paths. The clock period of a pipelined data path is proportional to the propagation delay of its

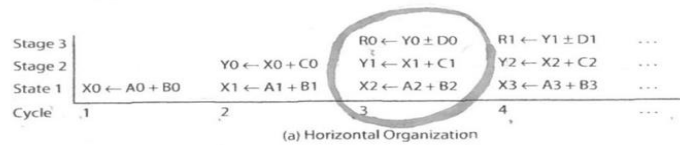


Fig 6.5

Pipeline chart for 3 stage pipeline

Refer to Figure 6.5 -Horizontal Pipeline chart (Cont)

- The pipeline chart in Fig. 6.5(a) has horizontal organization with clock cycles shown on the x-axis.
 - Pipeline chart illustrates various chart, does not include 1 cycle delay caused by interfacing registers
 - Has horizontal organization with clock cycles shown on x-axis.
 -

Sequential circuit: Pipeline performance

- In the example, R0 being the first result, requires 3 clock cycles to complete. (Ignore 1 cycle for interface registers)
- After R0, R1, R2.....Rn each output requires 1 clock cycle to produce results.
- Reduces time required to compute N final results.
- K stage (linear) pipeline requires K cycles for output.
- Equation Eq6.4 estimates total time (T_{pipeline}) required to process stream size of N using K stage pipeline

$$T_{\text{pipeline}} = K * \tau_p + (N-1) * \tau_p \quad (\text{Eq 6.4})$$

- Estimates total time- T_{single} -cycle to process data stream of size N using single cycle data path.

$$T_{\text{Single-cycle}} = N * K * \tau_p \quad (\text{Eq 6.5})$$

- Speed up is performance parameter that measures performance of faster system to a slower system.
- Defined as ratio of the time required by a slower system over faster system.
- Equation that defines speedup between a faster pipeline data path compared slower single cycle Datapath.

$$\text{Speedup} = T_{\text{single-cycle}} / T_{\text{pipeline}} = N * k * \tau / k * \tau + (n-1) * \tau \quad (\text{eq. 6.6})$$

- Efficiency is performance parameter that measures how well a system's resource's are utilized.
- Overall efficiency of a system is defined as the ratio of it's speedup to its maximum possible speedup.

- Efficiency = Speedup/K = $N / (k + N - 1)$
- As N approaches infinity, efficiency of pipeline approaches 100%.
- Throughput is performance parameter that measures a system's rate of processing
- Indicates the # of items (N) performed per second
- Eq. 6.8 defines the throughput of a linear pipeline with k stages

$$\text{Throughput} = N / T_{\text{pipeline}} = N / [(k \tau) + (N - 1) * \tau]$$