Pipelined architecture

Constraint: Data consistency, meaning the data going from input to output must go through the same number of pipeline register

Advantage: Increase in clock frequency, increase throughput, reduction in critical path(avoid setup violation)

Disadvantage: Greater utilization of logic resources

Timing analysis

Propagation delay: tpd= max delay from input to output

Contamination delay: tcd= min delay from input to output

Propagation delay: tpcq= time after clock edge that the output Qis guaranteed to be stable (i.e., to stop changing)

Contamination delay: tccq= time after clock edge that Q might be unstable (i.e., start changing)

tc > tpcq + tpd + tsetup

thold < tccq + tcd

tc > tpcq + tpd + tsetup + tskew

thold < tccq + tcd – tskew

­Fix hold violation: more buffer, reduce clock frequency, use better FF

Fix setup violation: reduce buffer, replace buffer with two inverters, pipelining

P(failure) = (T0/Tc ) e^(-(Tc - tsetup)/τ)

P(failure)/second = (NT0/Tc) e^((Tc - tsetup)/τ)

MTBF = 1/[P(failure)/second] = (Tc/NT0) e^((Tc - tsetup)/τ)

Data Required Time (setup) = Clock Arrival Time - Setup Time

Data Required Time (hold) = Clock Arrival Time + Hold Time

Slack = min<something> - max<something>

Setup = min DRT –maxDAT

Hold = min DAT –max DRT

$display similar to printf

$write similar to print

$time return current time

$monitor output data continuously

$strobe only output data at the end of simulation cycle