Minimum Average Latency (MAL)

- The previous slide shows all possible non-colliding operating modes.
- We can use the states C_0 and C_1 and transitions a_1, a_2, a_3 to describe different operating modes, e.g.
  - Cycle = $C_0a_1, C_1a_2, C_0a_1, C_1a_2, \ldots$
- Has minimum average latency (MAL)
- MAL = (2+3)/2 = 2.5 Periods/Datum
- This is the maximum performance obtainable from this pipe but it is an irregular cycle (i.e. t=2,3,2,3,...)

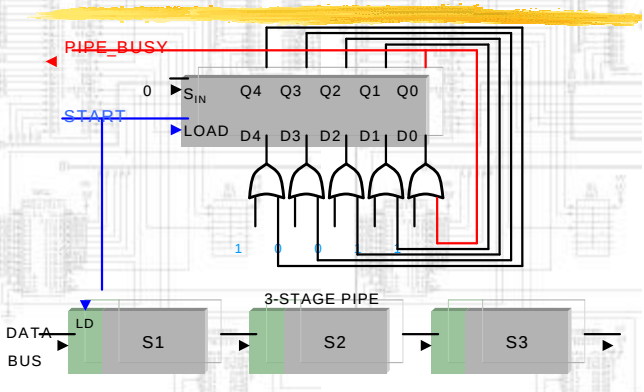
Minimum Latency

- The Cycle
  - Cycle = $C_0a_3, C_0a_3, C_0a_3, C_0a_3, \ldots$
- Has minimum average latency (MAL) = 3 and is regular
- This is termed the minimum latency

Implementation

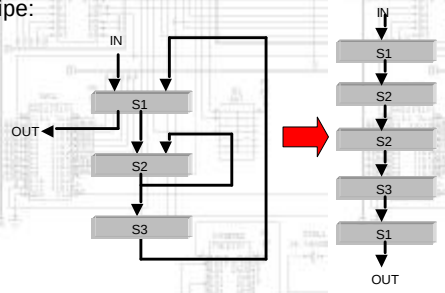
- The collision vector $C = (c_n, \ldots, c_1, c_0)$ provides the basis for access control to the pipeline. It is readily implemented with a PIPO shift register.

Implementation
PIPO



Re-circulating Pipe

- Of course any re-circulating pipe may be made non-circulating by duplication of its stages. E.g. for our example pipe:



Problem 3.1

- Assume that we have a pipeline with four stages that performs an add operation on two input numbers.
- The pipeline has a delay of two units of time for each stage.
- Also, ignore storage time, memory access time, time to set up the pipeline control circuit, and so on.
- It is necessary to sum 16 numbers by this pipeline.
- Describe a methode that gives the minimum possible time to perform the sum of the number.
- Write the total time required by our methode.

Problem 3.1 (More)

- We wish to compute:
  - $S=\sum_{i=0}^{15} a_i$
- In a 4-stage f.p. pipe with input registers A and B.
- We let A receive a_i and B accumulate the partial sums P_i
- Initially B must be cleared.
- When a_{16} has been fetched we collect the P_i in pairs in A and B until we have constructed S