Describing Pipelines

Madhav P. Desai

April 5, 2014

Two ways of describing pipelines

- with distinct threads.
- by pipelining a loop.

An example

- We want to compute $d(k) = (a(k) + b(k)) \times c(k)$ for a sequence of numbers.
- The system should accept a sequence of numbers $\{a(k), b(k), c(k)\}$ and produce the sequence $\{d(k)\}$.

As separate threads

```
void Stage_0()
{
  while(1)
  {
    float a = read_float32("a_pipe");
    float b = read_float32("b_pipe");
    float c = read_float32("c_pipe");
    write_float32("c_forward_pipe", c);
    write_float32("s0_result", a+b);
  }
}
void Stage_1()
{
  while(1)
  {
    float c = read_float32("c_forward_pipe");
    float aSb = read_float32("s0_result");
    write_float32("d_pipe", aSb * c);
```

In a single loop

```
void Daemon()
{
  while(1)
  {
    float a = read_float32("a_pipe");
    float b = read_float32("b_pipe");
    float c = read_float32("c_pipe");
    write_float32("d_pipe", (a+b)*c);
  }
}
```

Loop Pipelining

- Loop pipelining involves executing multiple iterations of a loop at the same time.
- While doing this, all dependencies must be taken care of.
  - Operation order.
  - Data dependency.
  - Memory accesses.
  - Pipe accesses.

Example