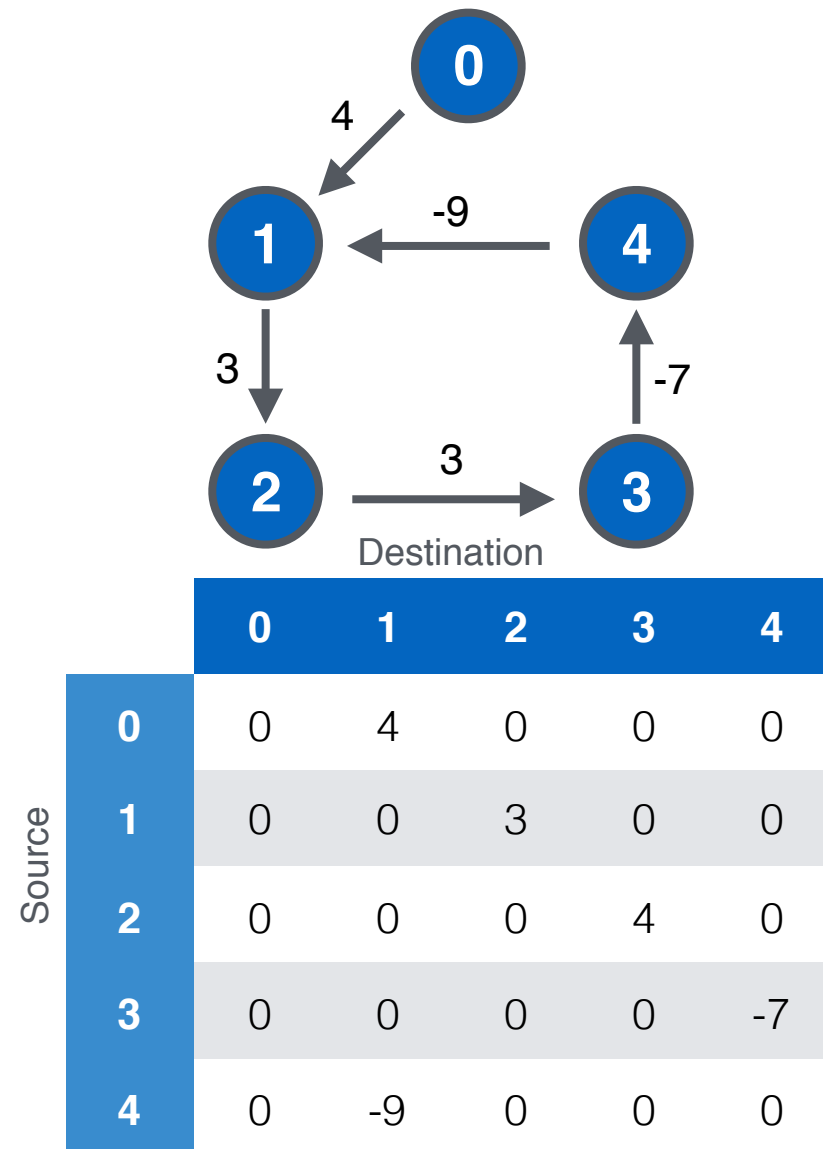


High-Frequency FOREX Trading: Identification of Triangular Arbitrage Opportunities

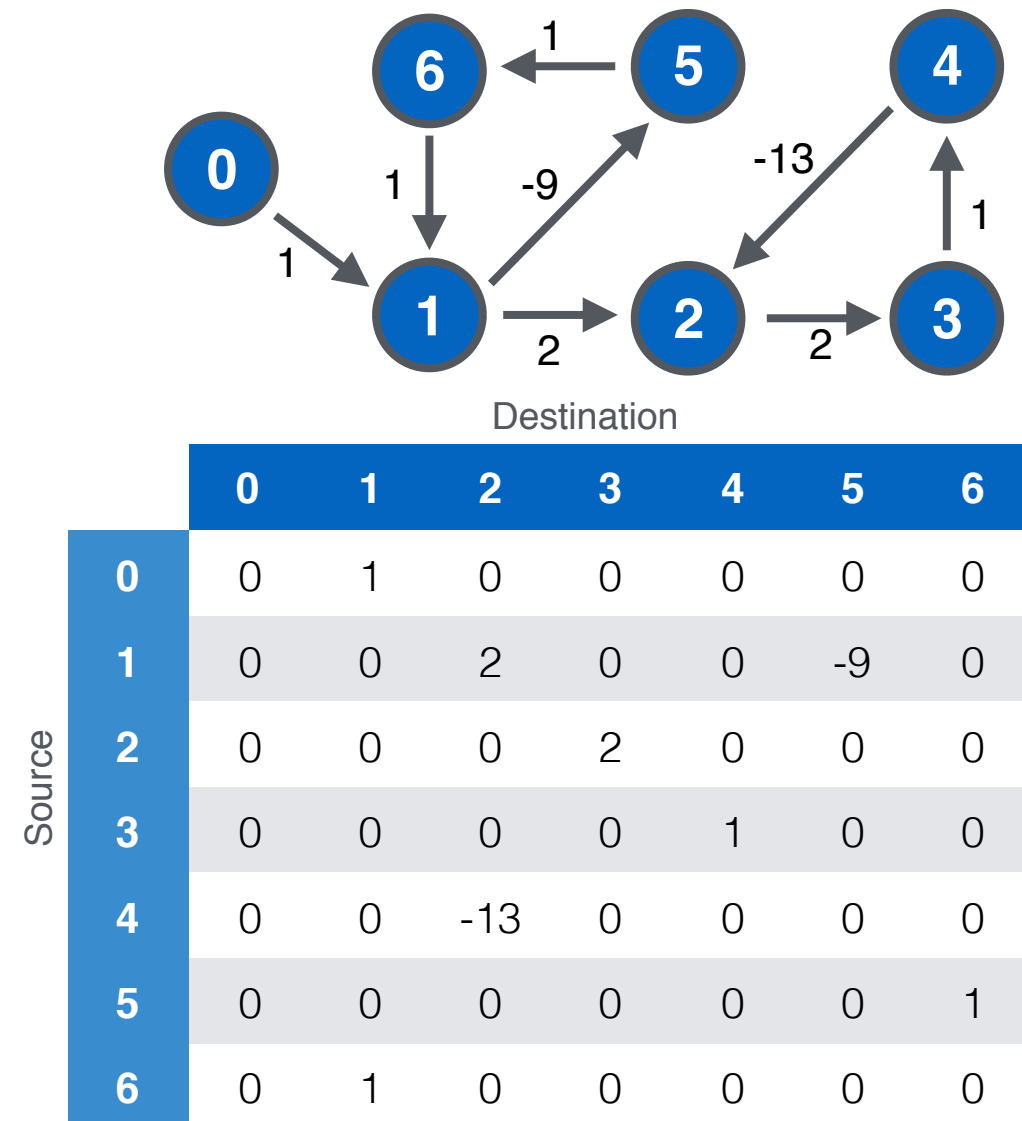
Graham Gobieski, Kevin Kwan, Ziyu Zhu, Shang Liu

Demos

Demo 1



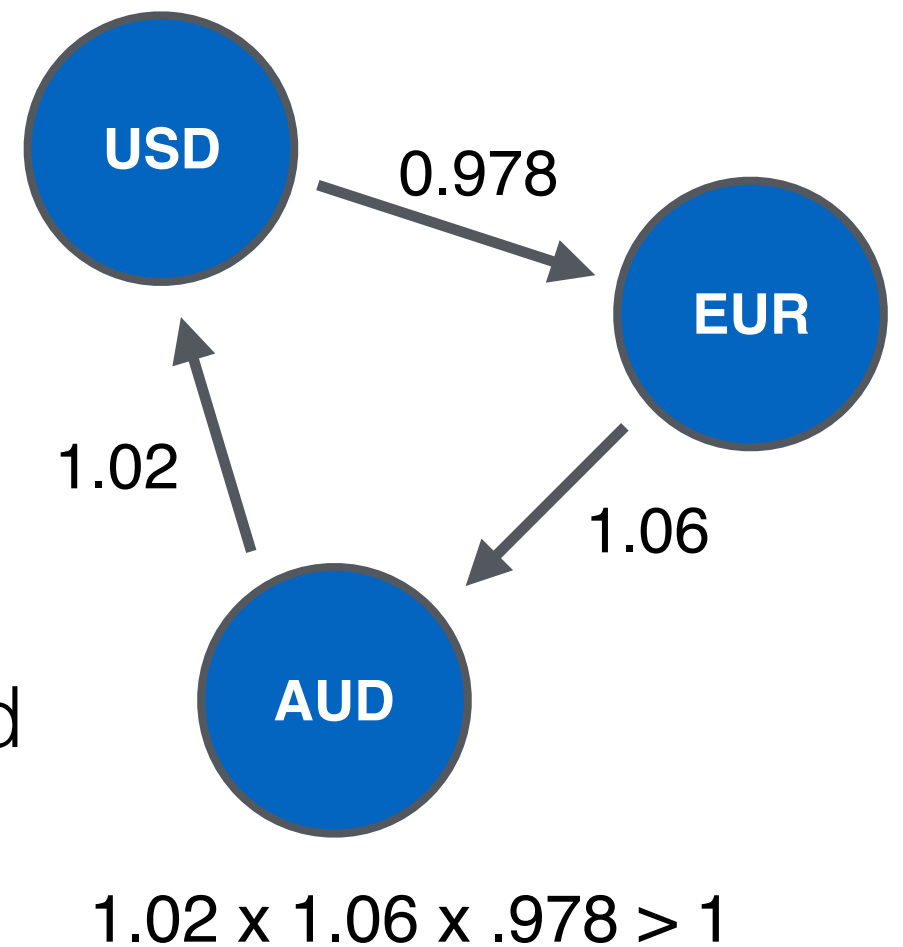
Demo 2



Demo 3: Live Cycles

Motivation

- High-Frequency Trading: taking advantage of opportunities (inefficiencies, etc.) on very short timescales
- Triangular Arbitrage: due to market inefficiencies, exchanging a currency between three or more currencies and arriving back at the original currency might be profitable
- Timescale: ~5-20ms, data streams over network



Bellman-Ford

```
for each vertex x in V do
  if x is source then
    w(x) = 0
  else
    w(x) = INFINITY
    p(x) = NULL
  end if
end for
```

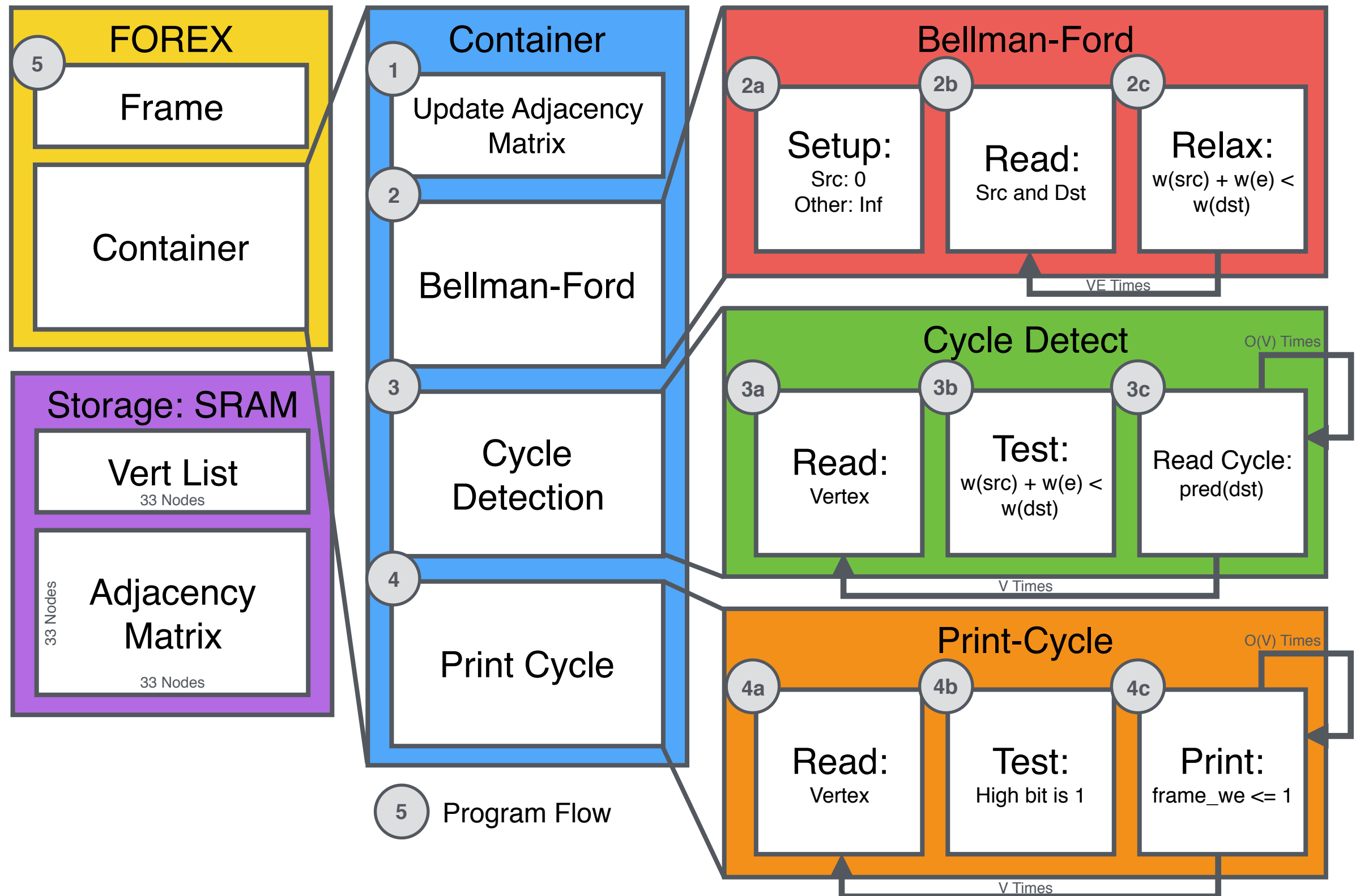
```
for i = 1 to v - 1 do
  for each edge(i, j) in E do
    if w(i) + w(i, j) < w(j) then //Relaxation
      w(j) = w(i) + w(i, j)
      p(j) = i
    end if
  end for
end for
```

```
for each edge(i, j) in E do
  if w(j) > w(i) + w(i, j) then
    //Found Negative-Weight Cycle
  end if
end for
```

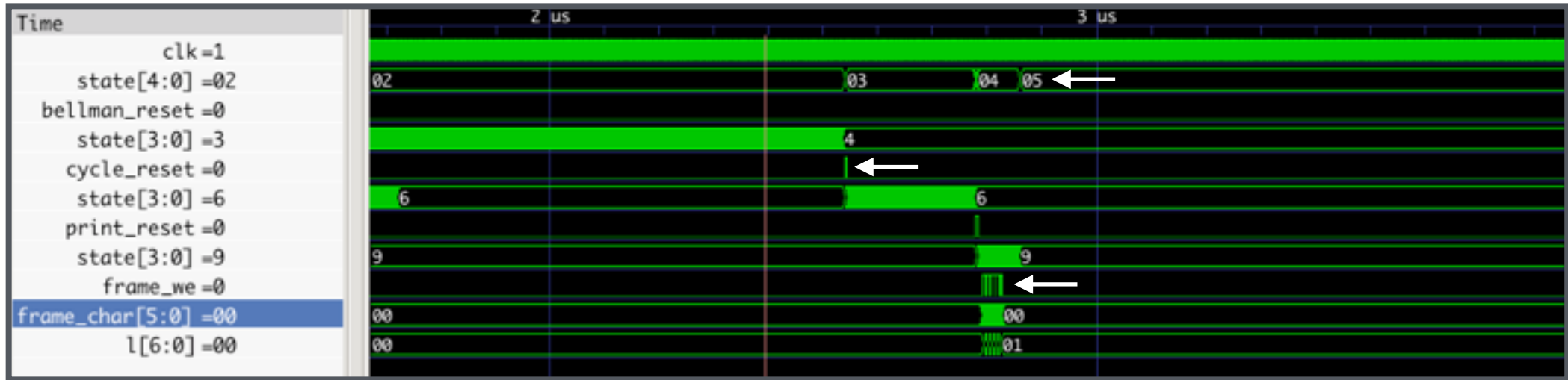
Transformation

1. $w_1 * w_2 * w_3 * \dots * w_n > 1$
2. $\log(w_1) + \log(w_2) + \log(w_3) + \dots + \log(w_n) < 0$
3. $-(\log(w_1) + \log(w_2) + \log(w_3) + \dots + \log(w_n)) < 0$

Hardware Design



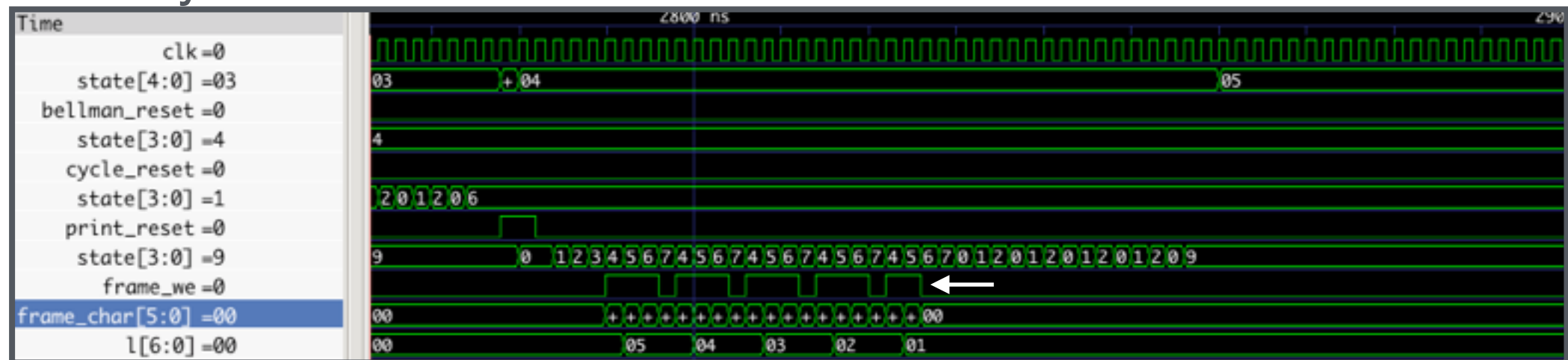
Overview



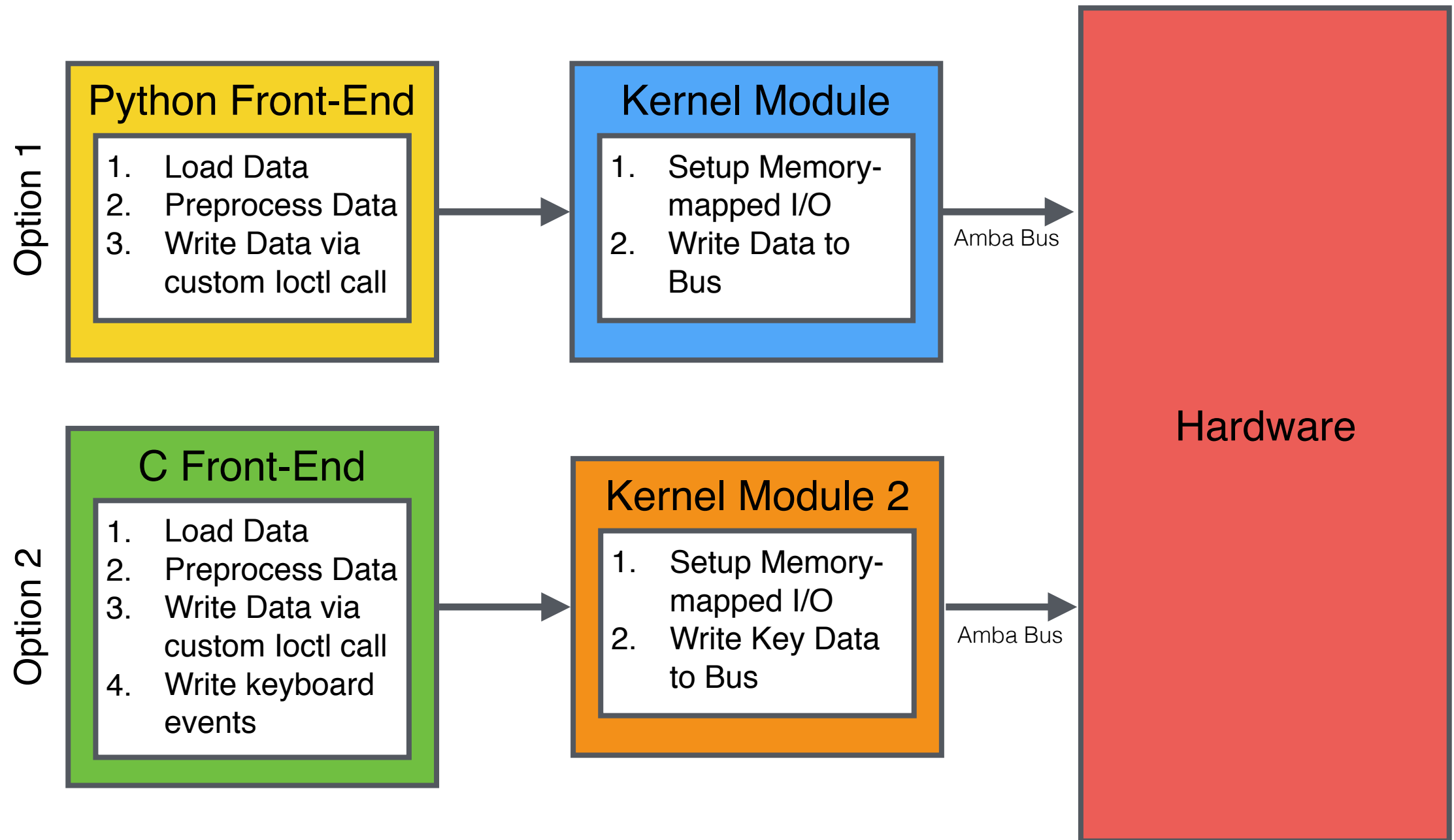
Bellman



Print Cycle



Software Design



Challenges

- Obviously Timing...
- Memory Accesses
- Nested Non-blocking Assignments
- Combinational Logic and Sequential Logic Working Together
- Compile Time