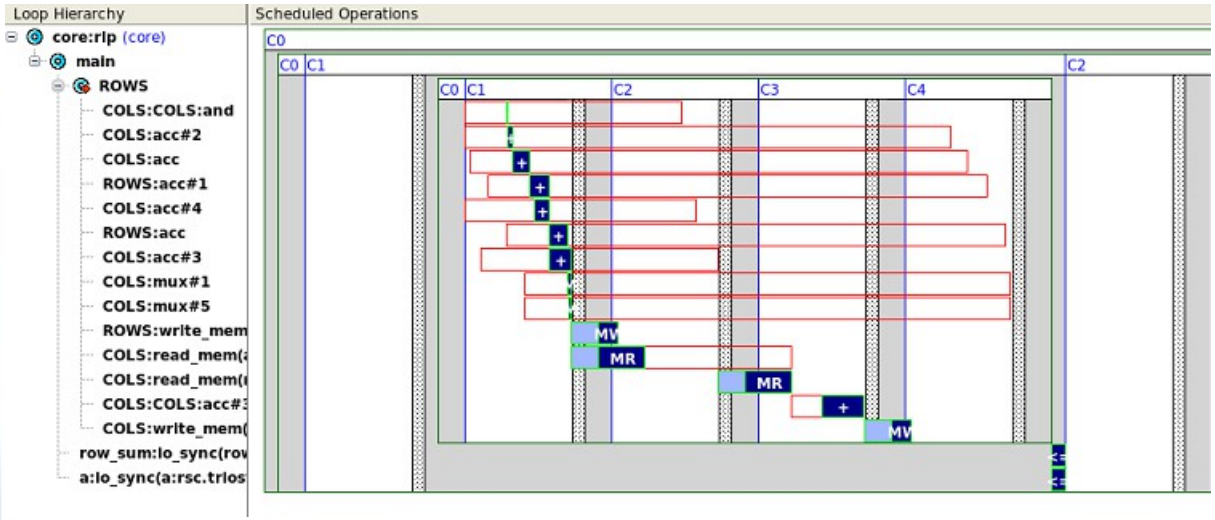
First Implementation

Best latency with II=3 can be seen in version .v5



Single port R/W memory means that we need 1 clk for one R/W operation. We can only pipeline up to 1RW/cycle.

Solution	Latency Cycles	Latency Time	Throughput Cycles	Throughput Time	Total Area	Slack
compute_row_sum.v7 (extract)	15	30.00	15	30.00	388.99	0.29
compute_row_sum.v6 (extract)	15	30.00	19	38.00	397.38	0.23
compute_row_sum.v5 (extract)	**43**	**86.00**	**47**	**94.00**	**357.46**	**0.28**

Minimizing the Initiation Interval

- We only need to write in the memory once to save the sum result.
- We do that by creating the local variable rows to save the intermediate results.

```
12  #pragma hls_design top
13  void CCS_BLOCK(compute_row_sum)(short a[N][M], short row_sum[N]) {
14    short rows;
15    ROWS:for (int i=0; i < N; i++) {
16      //row_sum[i] = 0;
17      rows=0;
18      COLS:for (int j=0; j < M; j++) {
19        //row_sum[i] += a[i][j];
20        rows +=a[i][j];
21      }
22      row_sum[i] = rows;
23    }
24  }
```