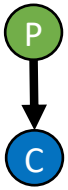
Original program

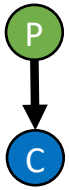
```
for j in [0, 5]
    M[j] = P(j)
for i in [0, 2]
    C(i) = M[2*i] + M[2*i + 1]
```

Data depence graph



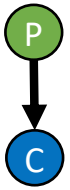
*M[2*i + 1] and*
*M[2*i] -> C(i)*

Schedule constraint graph



$S_C(i) >= S_P(2*i + 1)$

Assuming Rectangular Schedule



$q_C i + d_C \geq q_P(2 i + 1) + d_P$

Solution

$q_C = 2$
$q_P = 1$

$d_C = 1$
$d_p = 0$

New Loop Nest

```
for k in [0, 5]
  M[k] = P(k)
  if (k >= 1 && (k – 1) % 2 == 0):
    C((k – 1) / 2) =
      M[(k – 1)] + M[(k – 1) + 1]
```

Loop nest with stack memories

```
for k in [0, 5]
  M.push(P(k))
  if (k >= 1 && (k – 1) % 2 == 0):
    C((k – 1) / 2) =
      M.peek(-1) + M.peek(0)
```

Final Hardware Design