

```

mu unroll = 4 → 4 elements from A
nu unroll = 2 → 2 elements from B
latency = 3
ifetch = 4
nfetch = 1

```

```
/*Load A0..A2 regs, and B0..B1 regs */
```

```
temp1 = rA0 * rB0
temp2 = rA1 * rB0
temp3 = rA2 * rB0
```

} Start of the pipeline
latency = 3

```
/* Unroll loop */
rC1 = rC1 + temp1
temp1 = rA3 * rB0
rC2 = rC2 + temp2
temp2 = rA0 * rB1
rC3 = rC3 + temp3
temp3 = rA1 * rB1
rC4 = rC4 + temp1
temp1 = rA2 * rB1
rC5 = rC5 + temp2
temp2 = rA3 * rB1
/*Load A0..A2 regs, and B0..B1 regs */
```

```
rC6 = rC6 + temp3
temp3 = rA0 * rB0
rC1 = rC1 + temp1
temp1 = rA1 * rB0
```

```
rA0 = *pA0;
rB0 = *pB0;
rA1 = pA[1];
rA2 = pA[2]
```

} ifetch (4)
- first rA0, rB0
- then rA0 while < nfetch
- then rB0 while < nfetch

```
temp1 = rA0 * rB0
rA3 = pA[3] ← nfetch loads (first A regs)
temp2 = rA1 * rB0
rB1 = pB[40] ← nfetch loads
temp3 = rA2 * rB0 ← nfetch loads?
```

```
/* Unroll loop */
rC1 = rC1 + temp1
temp1 = rA3 * rB0 ← nfetch loads?
rC2 = rC2 + temp2
temp2 = rA0 * rB1
rC3 = rC3 + temp3
temp3 = rA1 * rB1
rC4 = rC4 + temp1
temp1 = rA2 * rB1
rC5 = rC5 + temp2
temp2 = rA3 * rB1
```

```
rA0 = pA[];
rB0 = pB[];
rA1 = pA[];
rA2 = pA[]
rA3 = pA[]
```

} ifetch + nfetch (4 + 1)
- first rA0, rB0
- then rA0 while < (ifetch+nfetch)
- then rB0 while < (ifetch+nfetch)

```
rC6 = rC6 + temp3
temp3 = rA0 * rB0
rB1 = pB[] ← nfetch loads
rC1 = rC1 + temp1
temp1 = rA1 * rB0 ← nfetch loads?
```