

# Optimizing Recovery Logic in Speculative High-Level Synthesis

Dylan Leothaud\*, Jean-Michel Gorius\*, Simon Rokicki\*, Steven Derrien<sup>†</sup>

\*Univ Rennes, Inria, CNRS, IRISA

<sup>†</sup>UBO, Lab-STICC

This work is partially funded by the French National Research Agency (ANR) as part of  
the LOTR project



THE CHIPS  
TO SYSTEMS  
CONFERENCE

anr®

SPONSORED BY

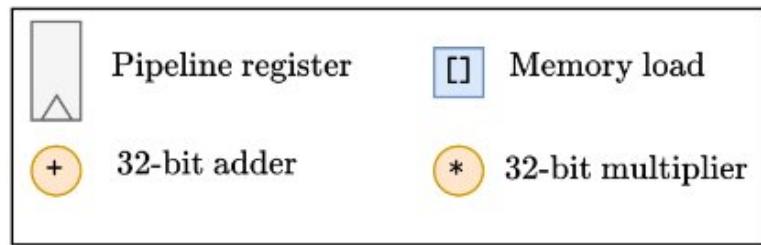
CEDA®  
IEEE Council on Electronic Design Automation

SIG  
acm da

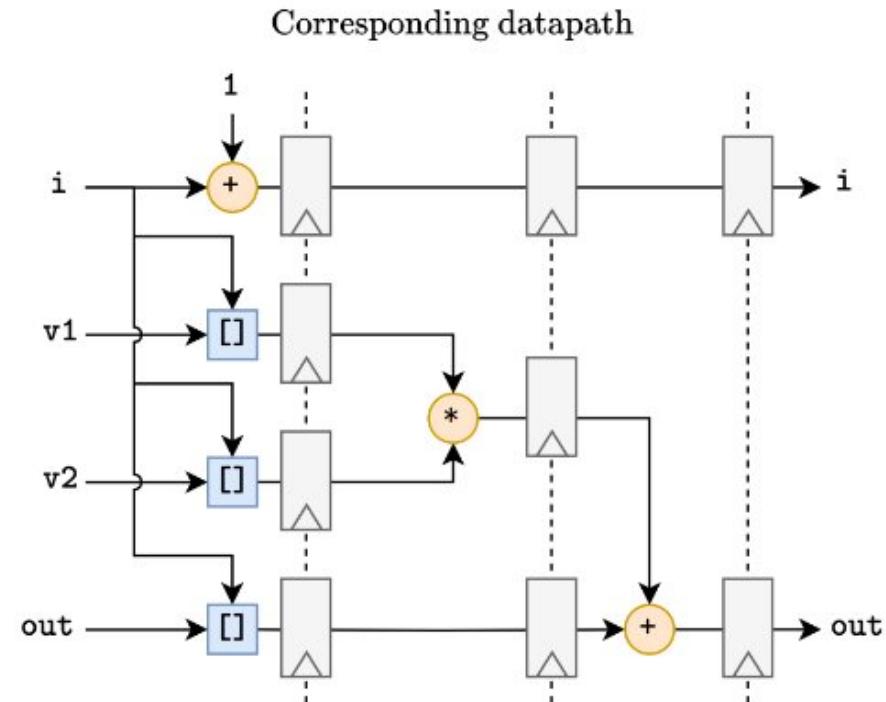
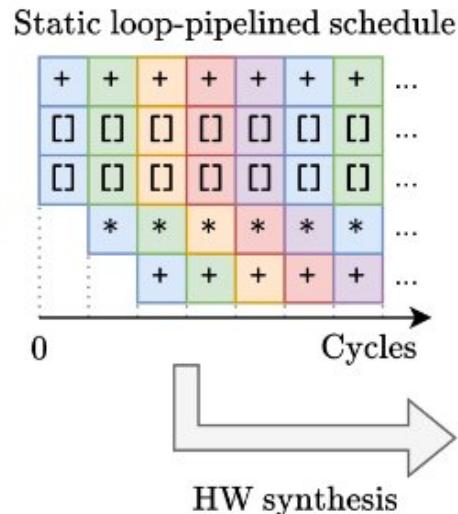
# High-Level Synthesis

High-Level Synthesis (HLS) tools generate HDL from  
C++ description

```
int vec_mul(int *v1, int *v2,
            int out, int n) {
    for (int i = 0; i < n; ++i)
        out += v1[i] * v2[i];
    return out;
}
```



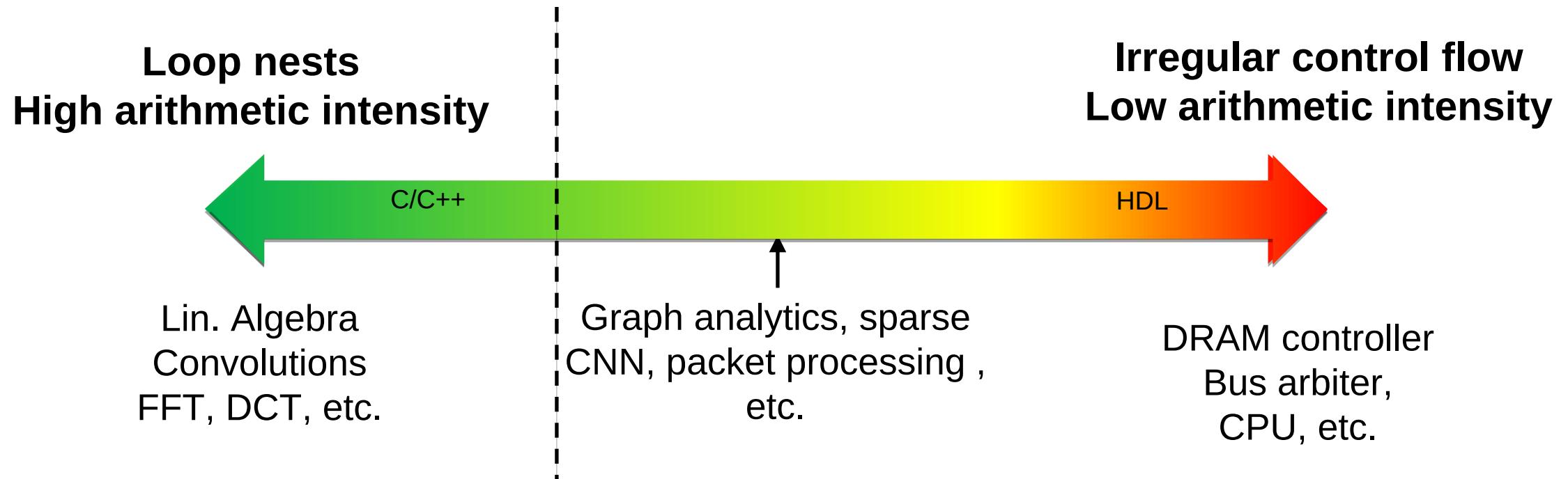
HLS



HLS automatically generates pipelined hardware

FPGA needs deep pipelining to be efficient

# What can I use HLS for in 2025

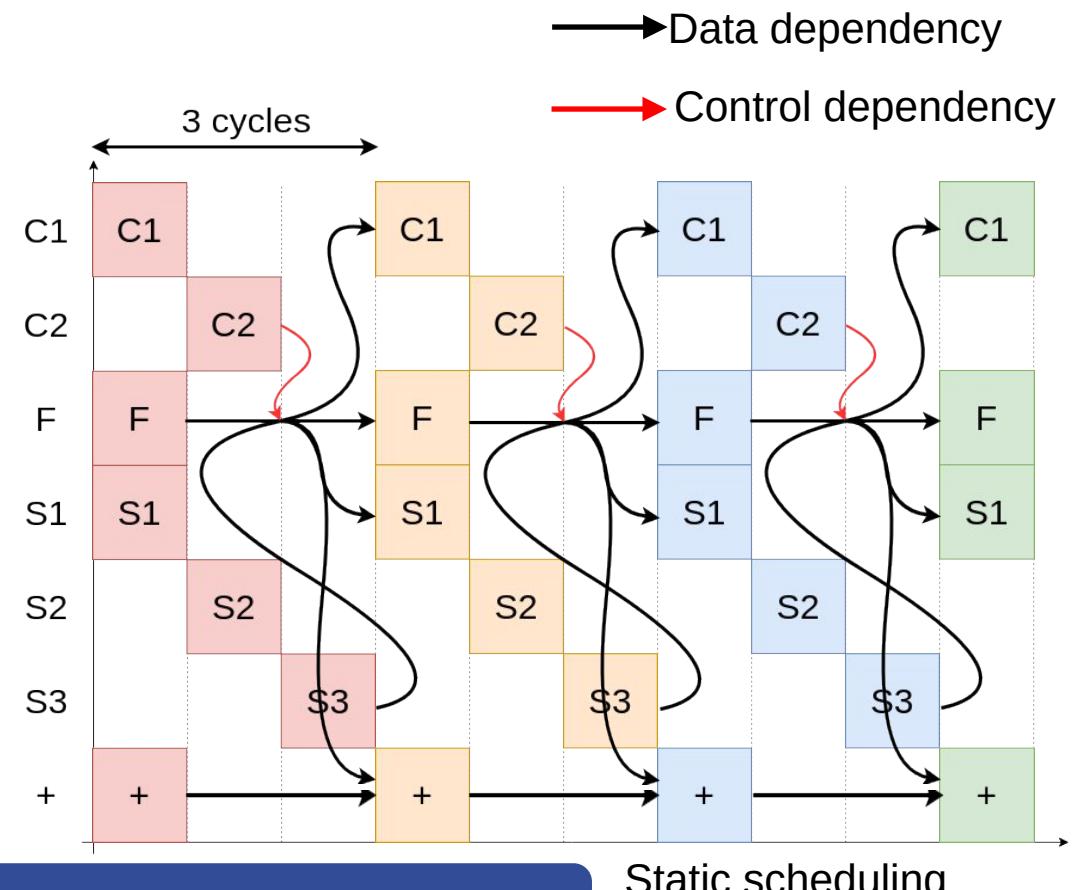
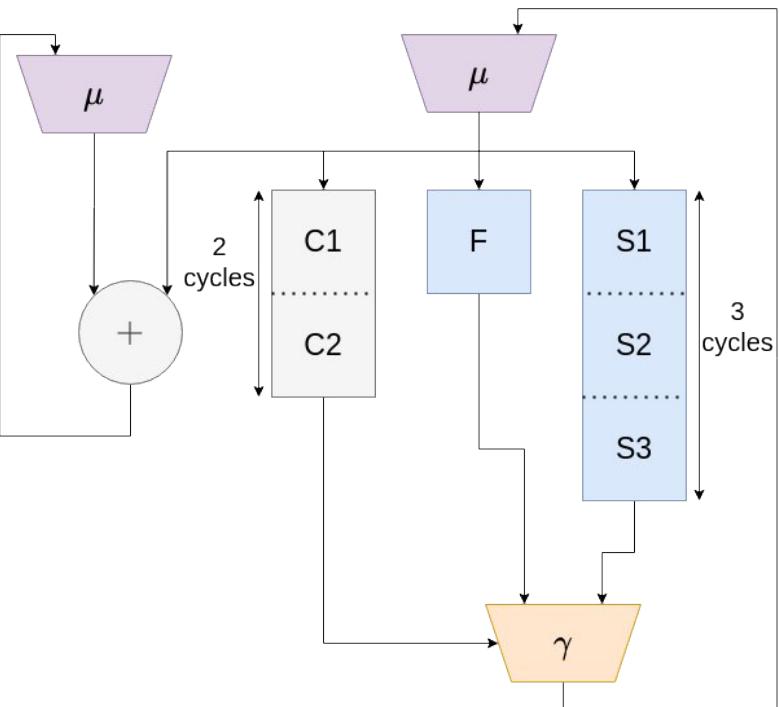


HLS tools struggle at efficiently scheduling kernels with irregular control-flow

# Static scheduling

SOTA commercial HLS tools all rely on static scheduling

```
while (1) {  
    y = y + x;  
    // 2 cycles  
    if (C(x)) {  
        // 1 cycle  
        x = F(x);  
    } else {  
        // 3 cycles  
        x = S(x);  
    }  
}
```

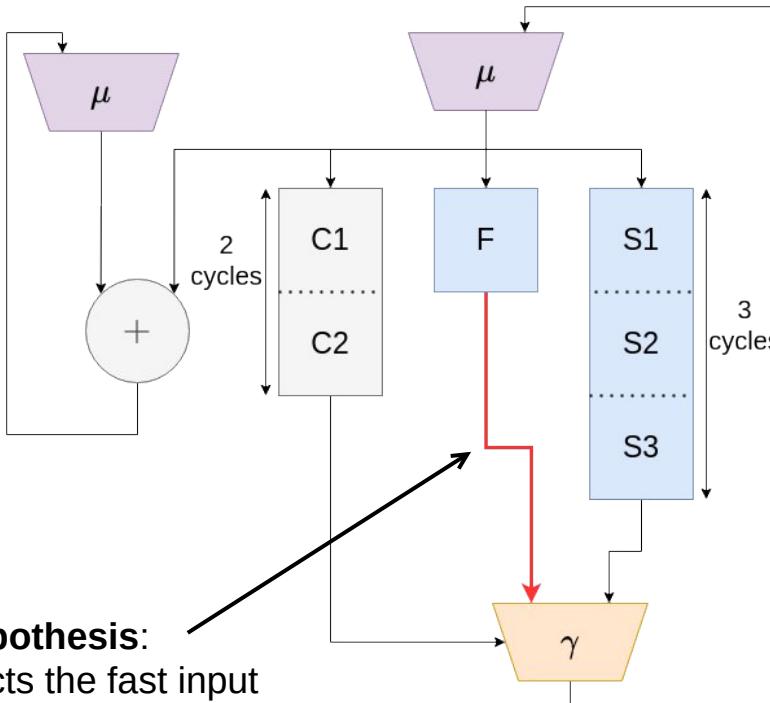


Static scheduling is pessimistic: based on worst case behavior

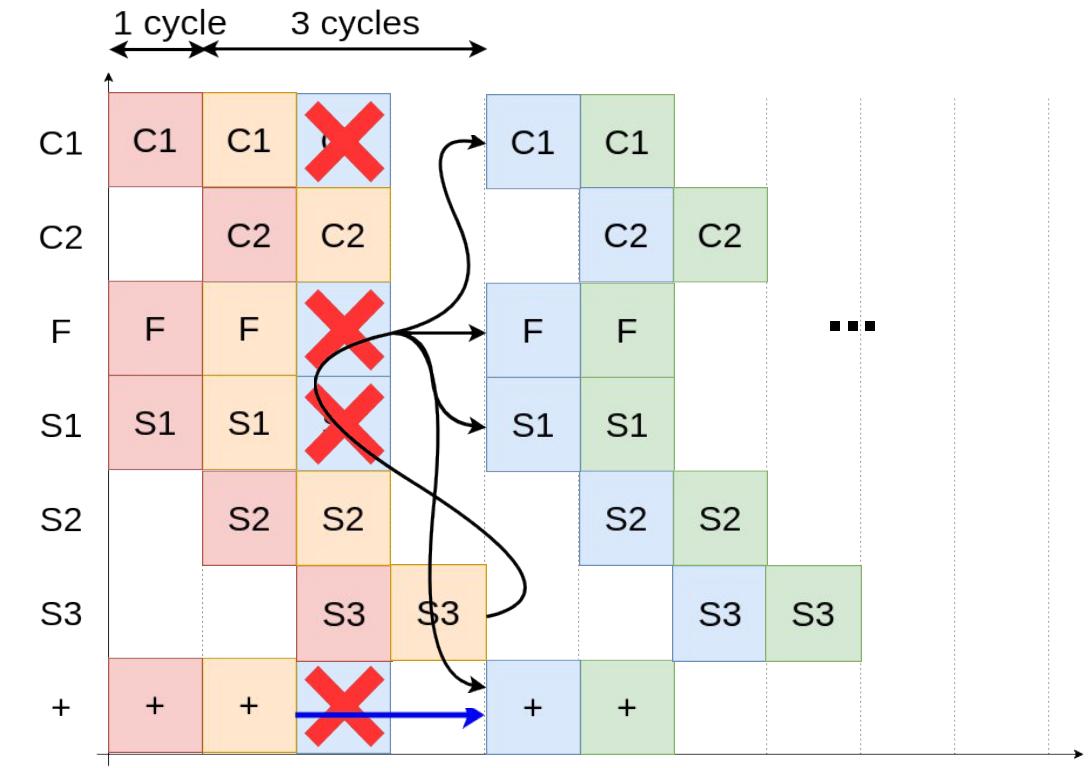
# Speculative scheduling

Speculative scheduling starts future iterations by predicting control-flow decisions

```
while (1) {  
    y = y + x;  
    // 2 cycles  
    if (C(x)) {  
        // 1 cycle  
        x = F(x);  
    } else {  
        // 3 cycles  
        x = S(x);  
    }  
}
```



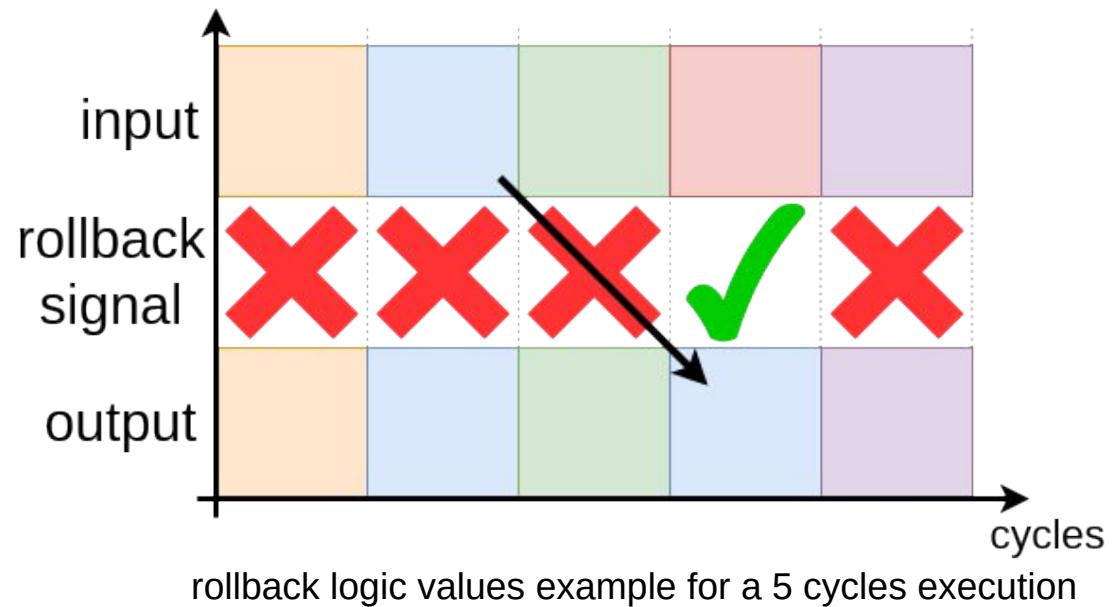
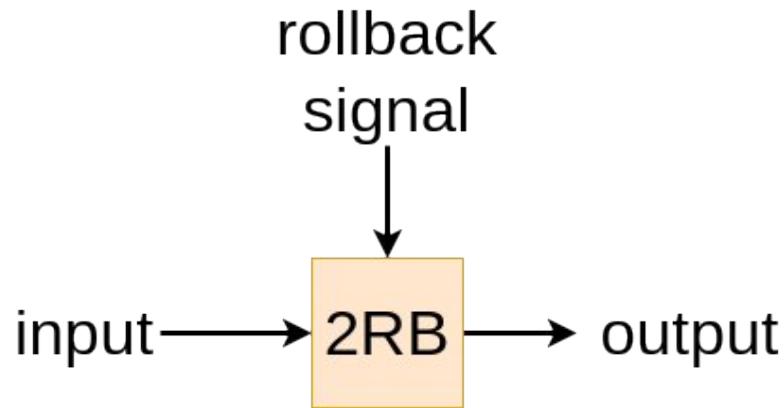
Speculation hypothesis:  
the gamma node selects the fast input



Some value need to be restored

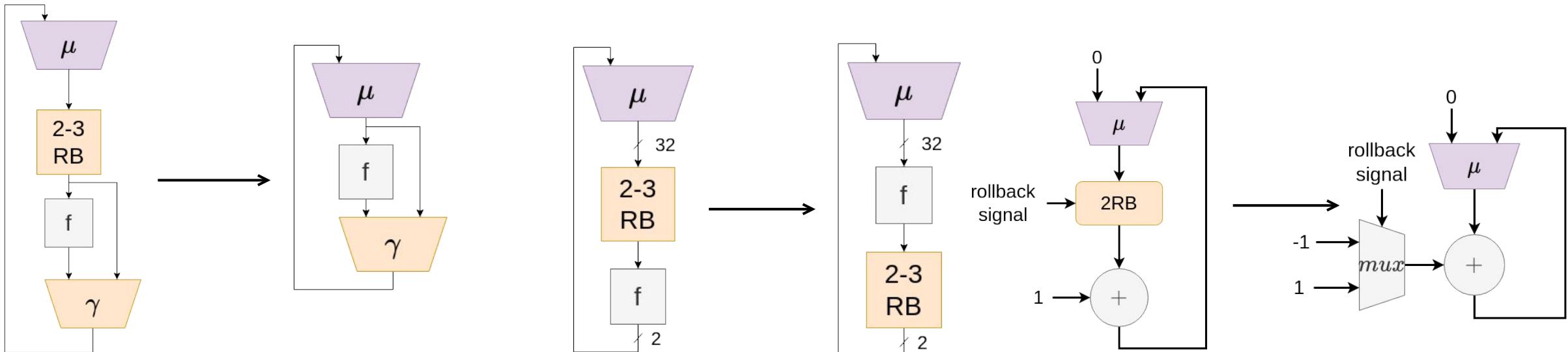


# Rollback logic behavior



Rollback logics are inserted after every  $\mu$  and  $\gamma$  node

# How to optimize rollback logic cost ?



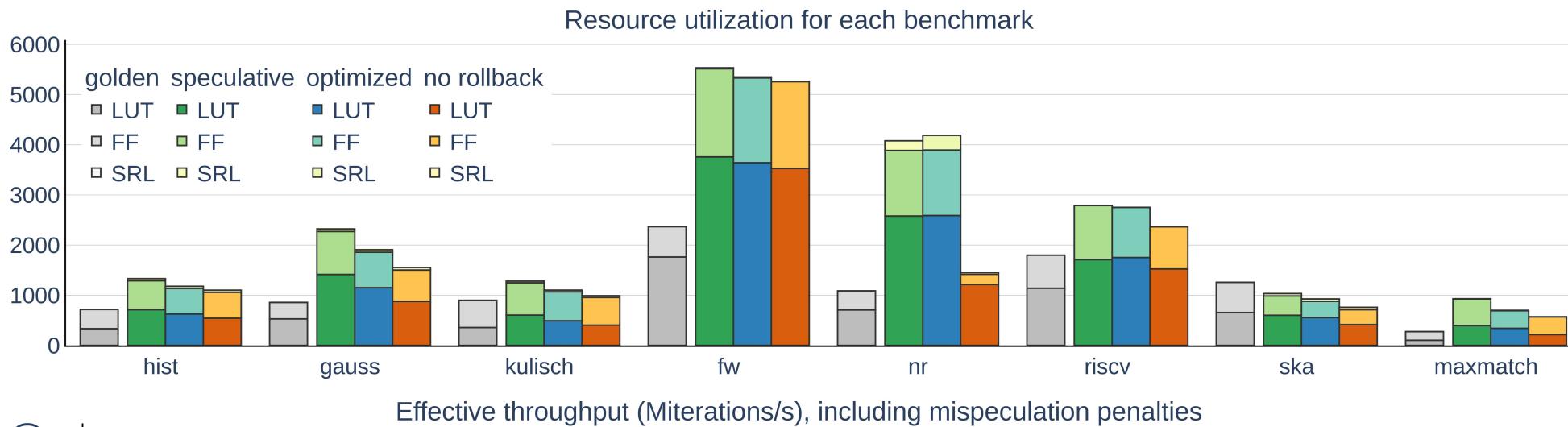
Remove rollback logic that can be proven unused

Retime rollback logics to minimal bitwidth values

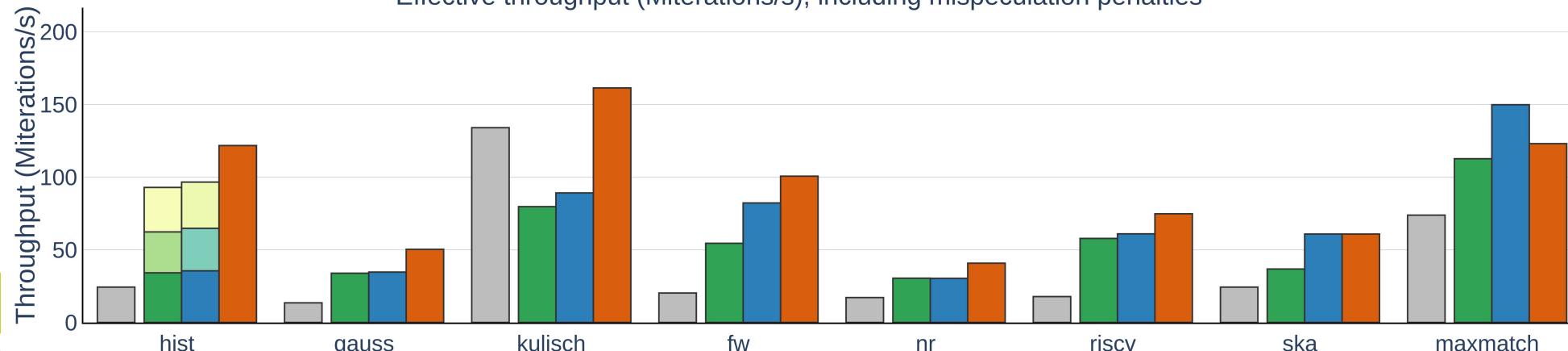
Reuse reversible operations' operators

# Experimental results

10%  
area  
reduction



13%  
throughput  
improvement



# Conclusion

Speculation opens up new opportunities for High-Level Synthesis

One challenge is to minimize rollback logic overhead

With our approach, it is possible to decrease the speculative circuit area and increase the clock frequency

