

# Optimizing Recovery Logic in Speculative High-Level Synthesis

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

<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

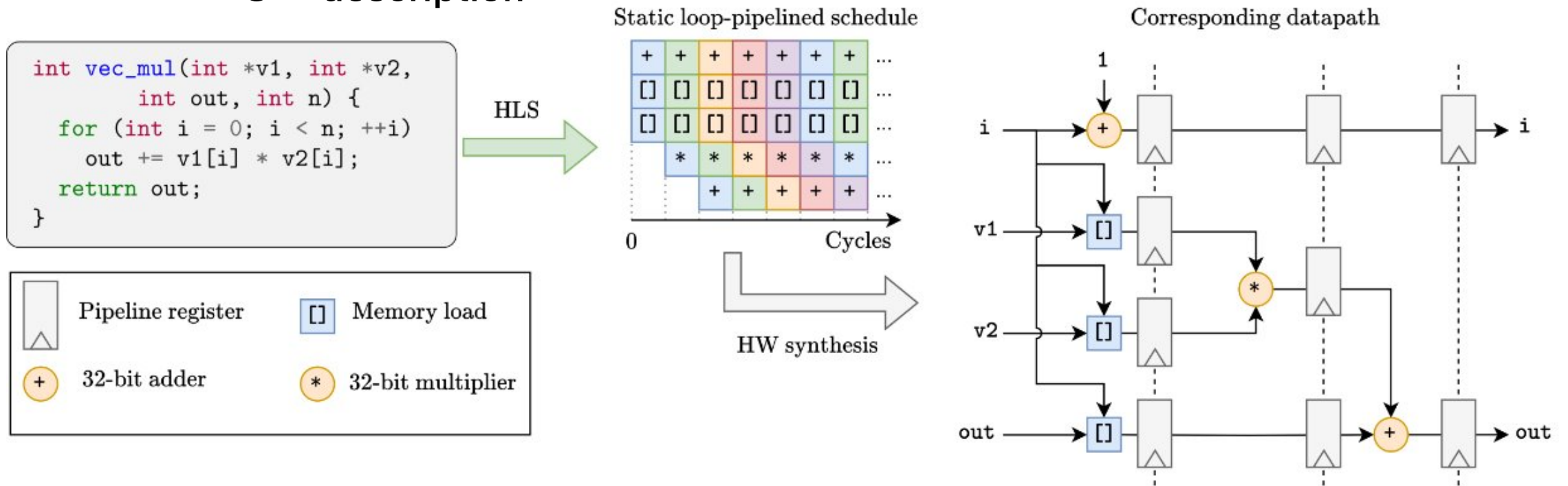


SPONSORED BY



# High-Level Synthesis

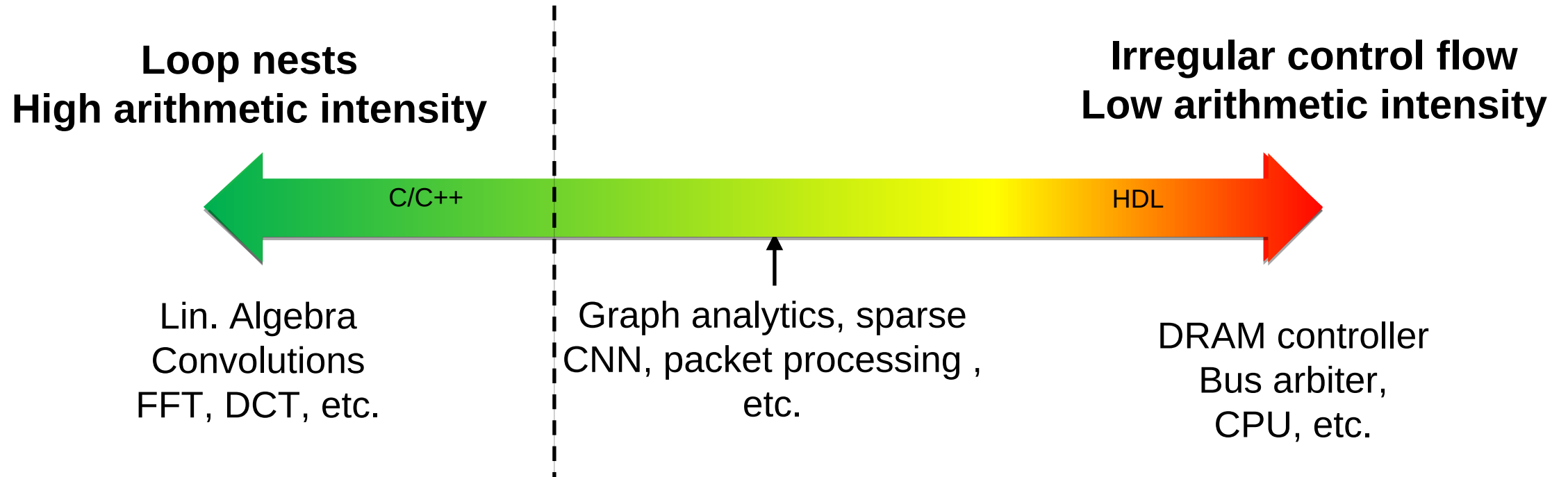
High-Level Synthesis (HLS) tools generate HDL from C++ description



HLS automatically generates pipelined hardware

FPGA needs deep pipelining to be efficient

# What can I use HLS for in 2025

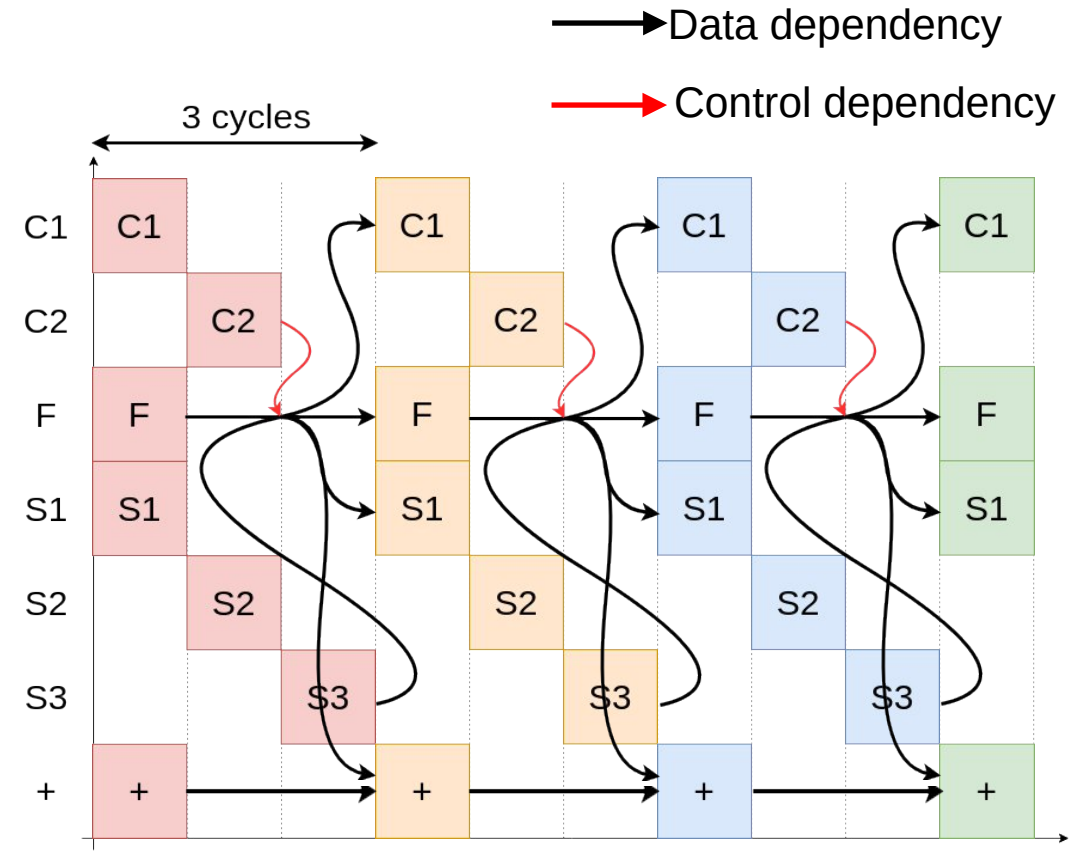
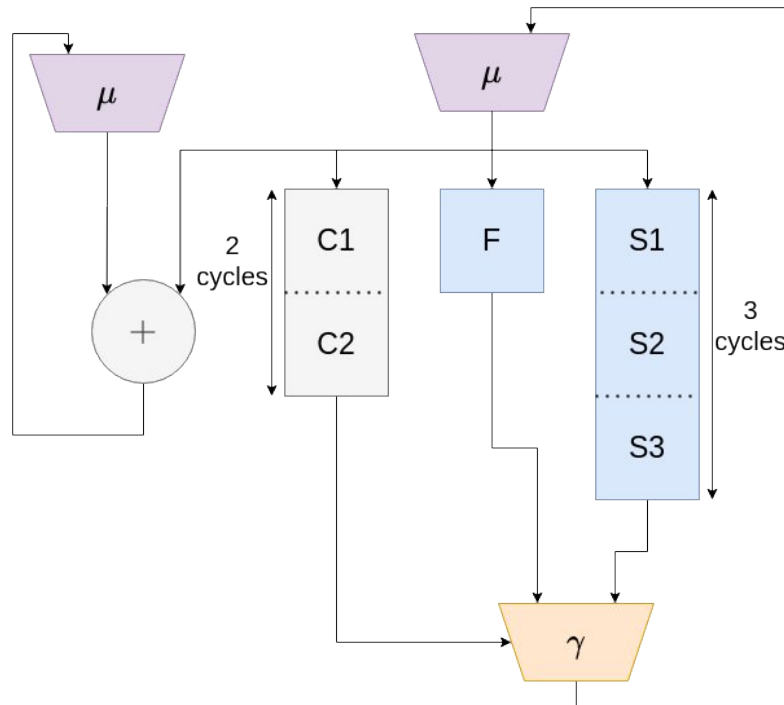


HLS tools struggle at efficiently scheduling kernels with irregular control-flow

# 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);  
  }  
}
```

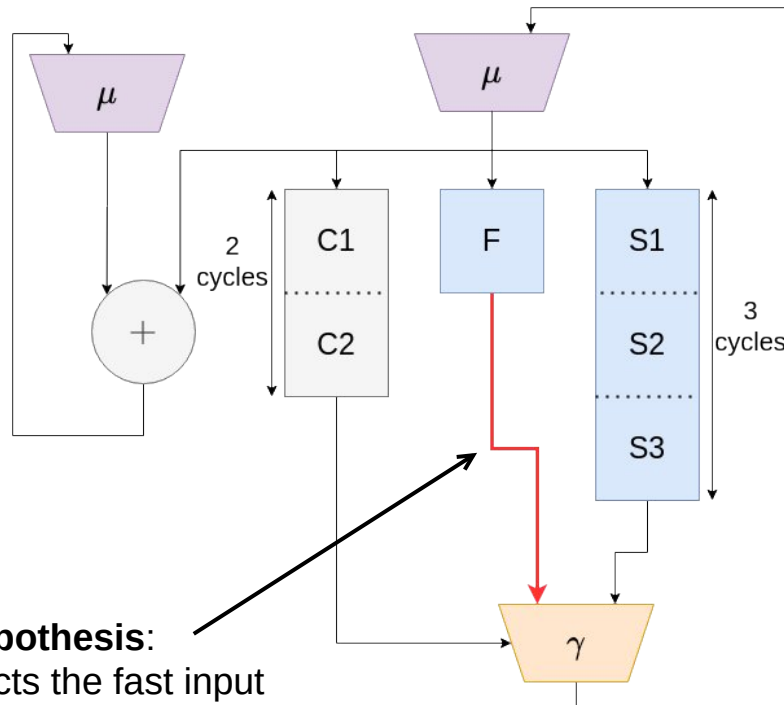


Static scheduling is pessimistic: based on worst case behavior

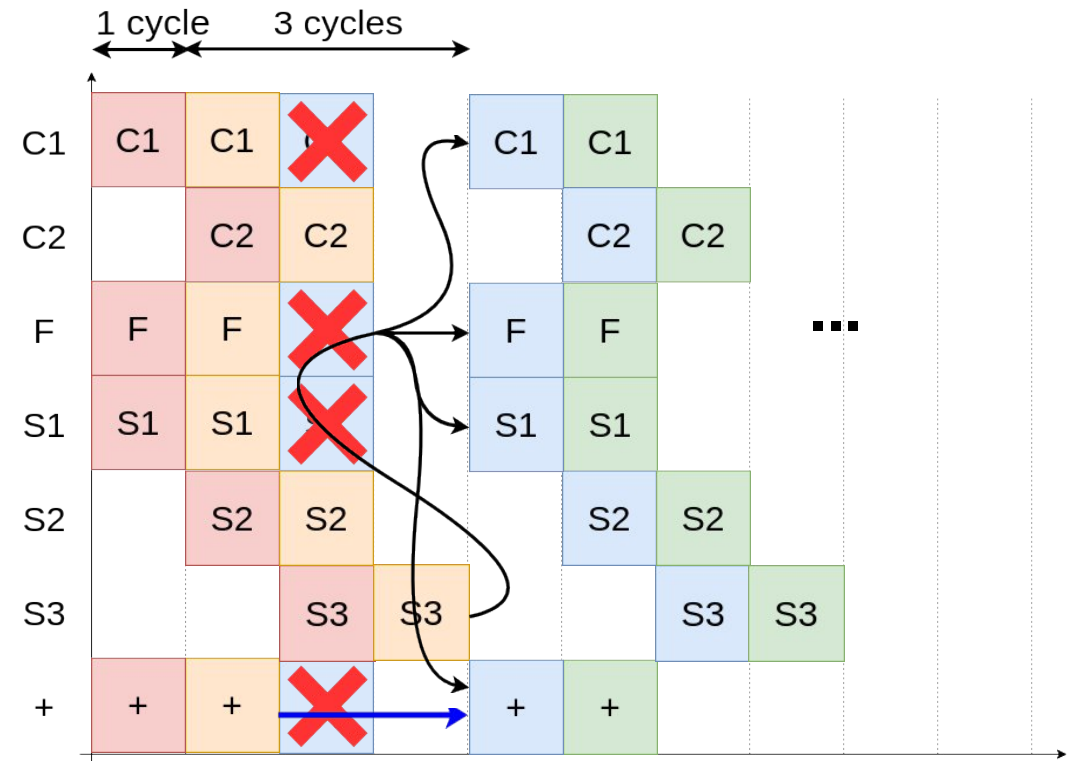
# Speculative scheduling

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

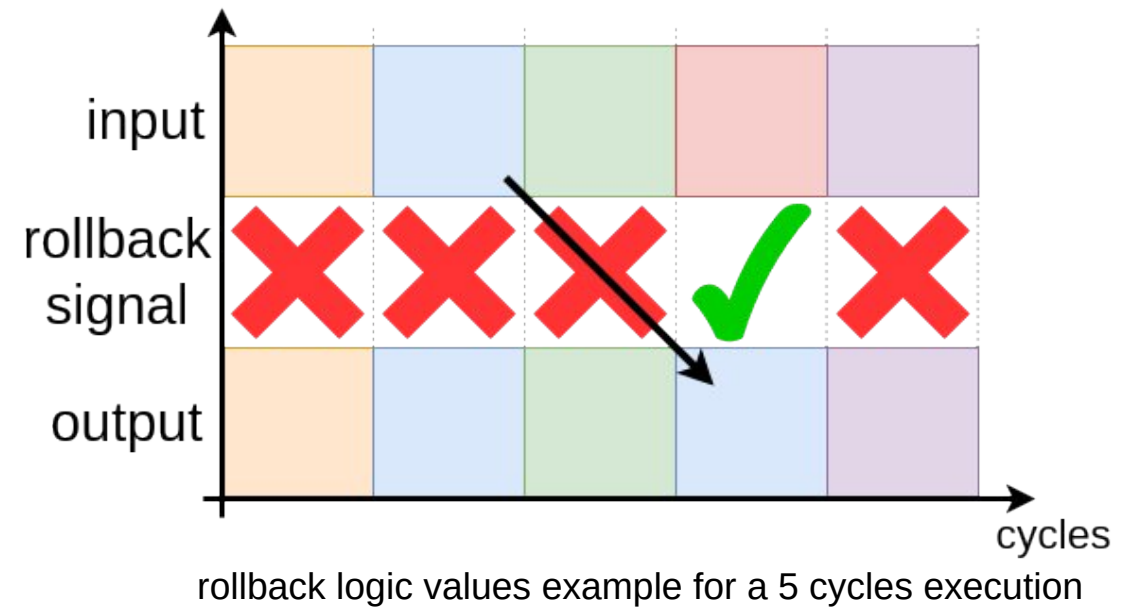
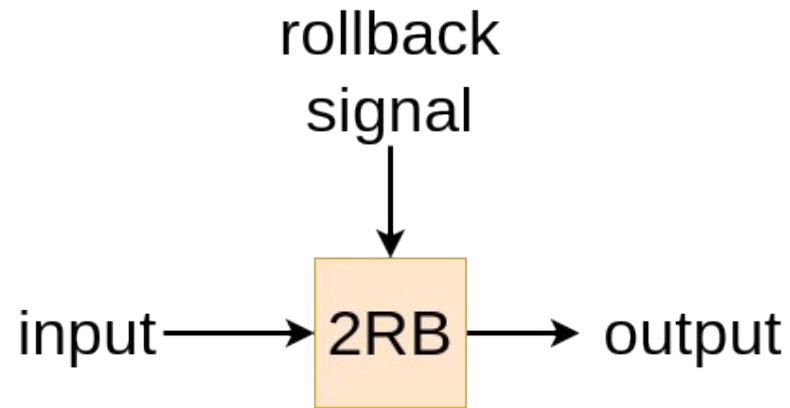


Some value need to be restored

Speculative  
scheduling

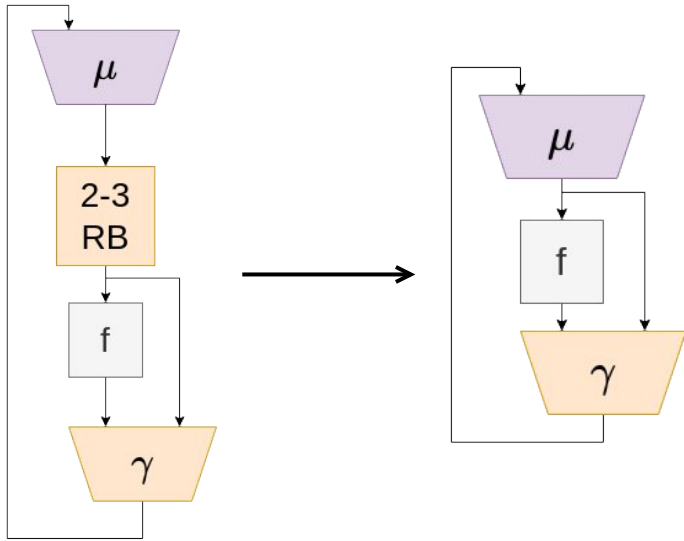


# Rollback logic behavior

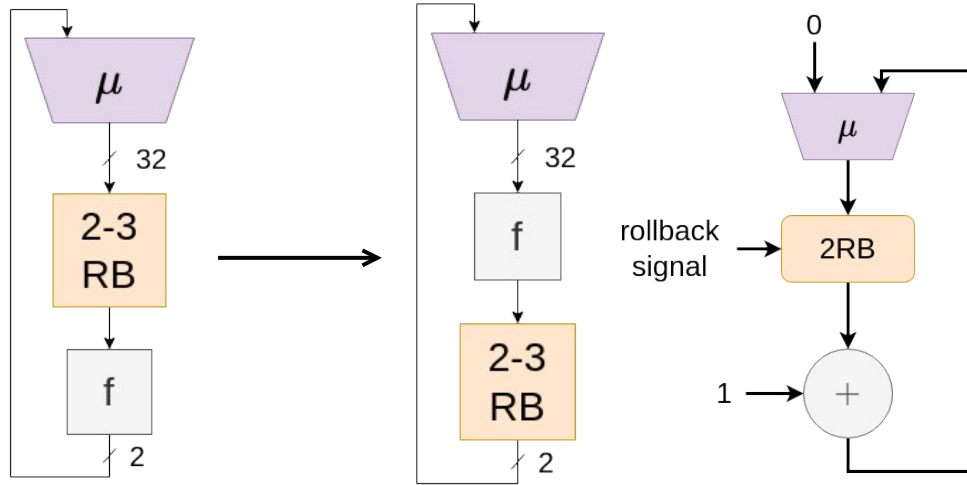


Rollback logics are inserted after every  $\mu$  and  $\gamma$  node

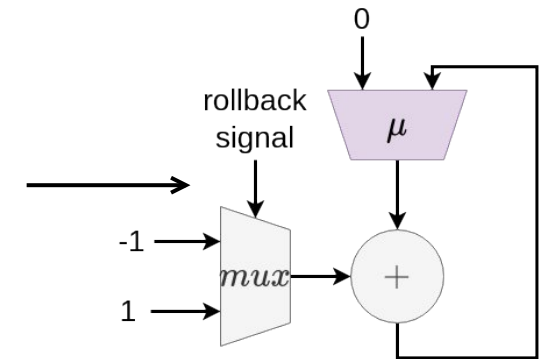
# How to optimize rollback logic cost ?



Remove rollback logic that can be proven unused

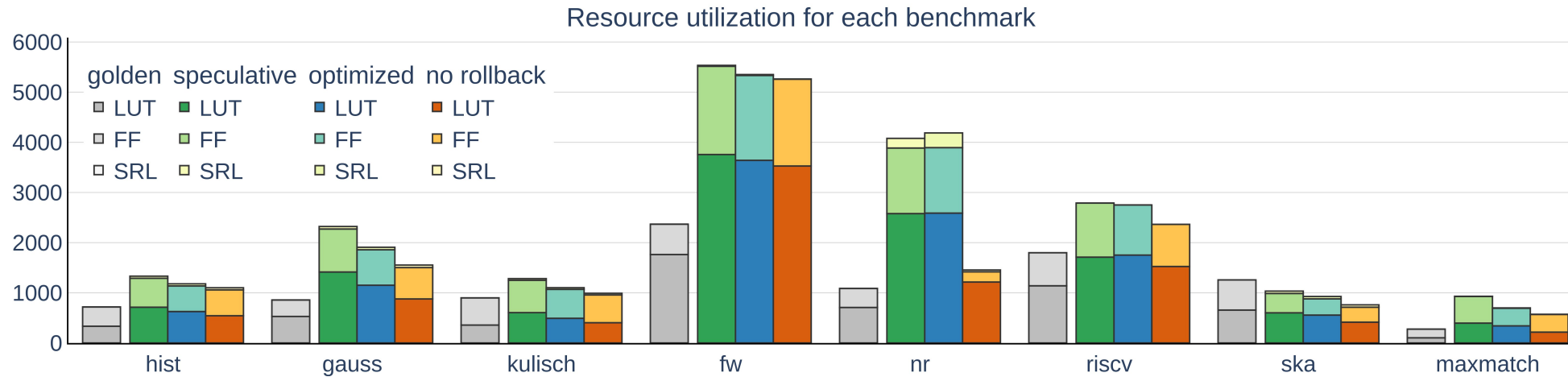


Retime rollback logics to minimal bitwidth values

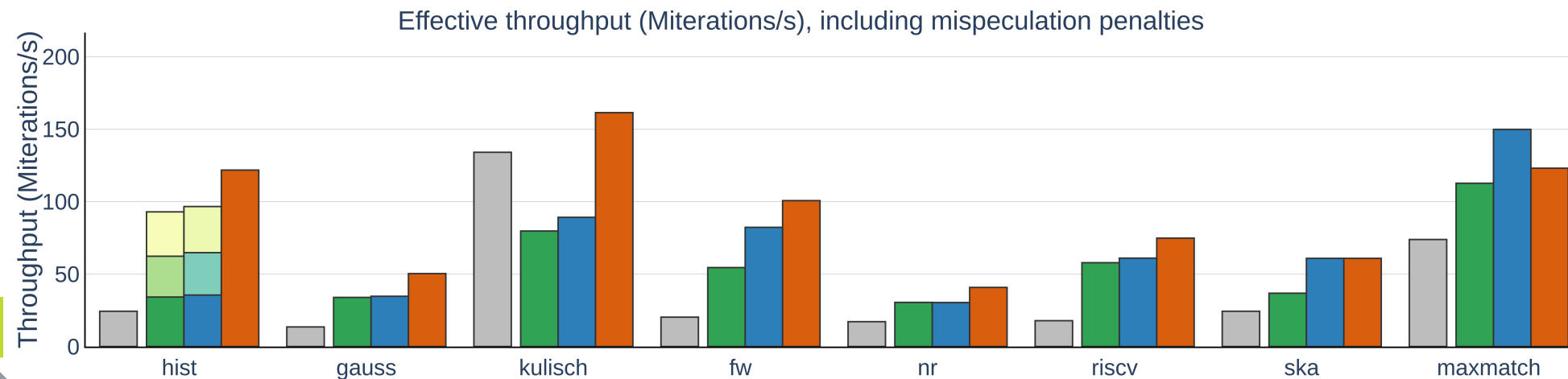


Reuse reversible operations' operators

# 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