Report 14.01.21

12/01/2020

Matteo Perotti
Luca Bertaccini
Pasquale Davide Schiavone
Stefan Mach

Professor Luca Benini
Integrated Systems Laboratory
ETH Zürich

Zfinx compiler

- Found and reported a bug → The riscv binutils repo was updated
- The compiler worked correctly with -O2 as optimization level
- Found another issue (in some corner cases) when using -O3 (-O2 -fpeel-loops)
- I reported the new bug and provided a simple C code example for which the compiler reports an error

Zfinx compiler

```
#define N 9
float matA[N*N];
float matB[N*N];
float matC[N*N];

int main() {
  for (int i = 0; i < N; ++i) {
    for (int j = 0; j < N; ++j) {
        matC[i*N+j] = matA[i*N] * matB[N+j];
    }
  }
  return 0;
}
```

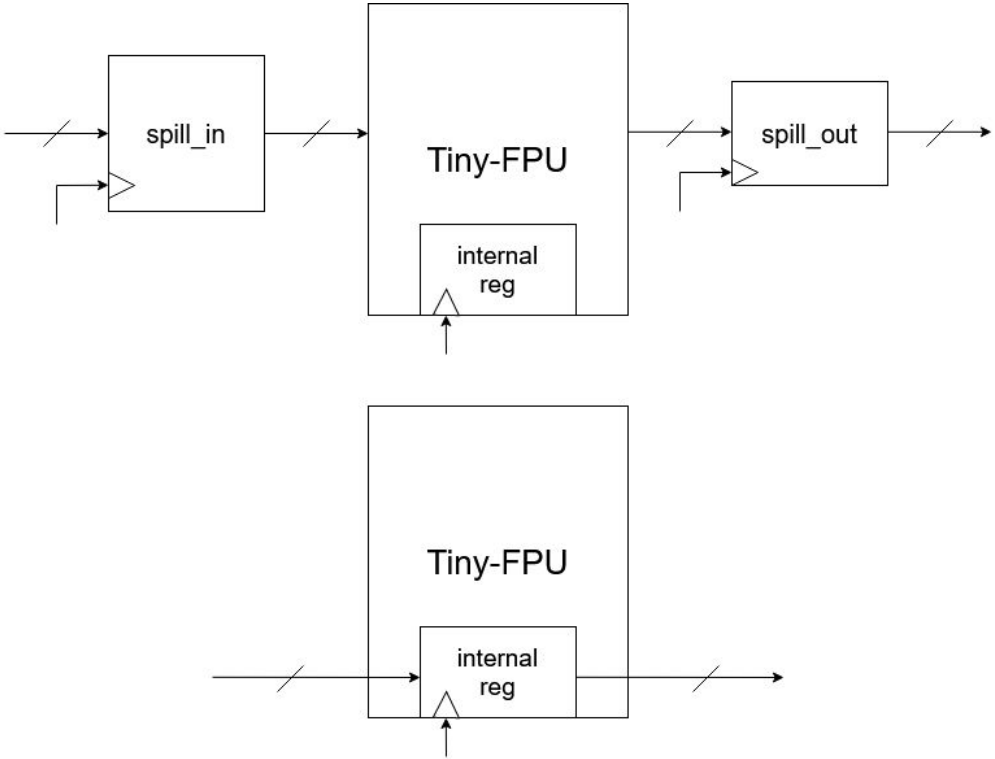
- No error if the matrices are defined as int
- No error if the matrices are defined inside the main

Further improvements

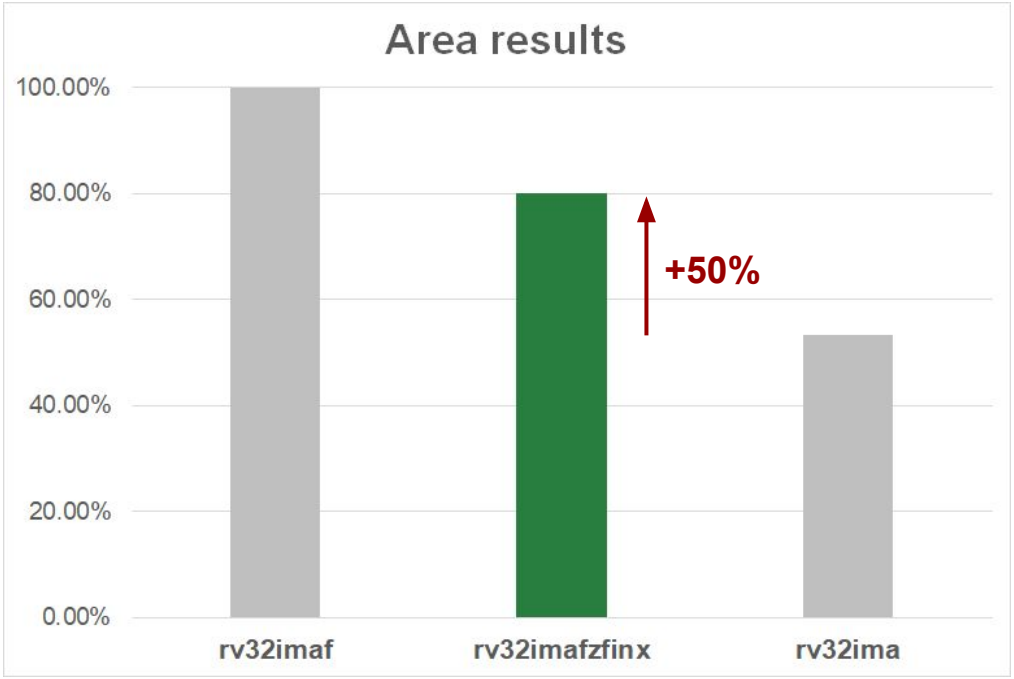
- In the FP subsystem, spill registers are wrapped around the FPU to cut the combinational path
- If out FPU is TinyFPU, these registers are a duplication of TinyFPU internal registers
- We could remove the input spill registers already (25% of the area of FP32 Tiny-FPU) → ~1.43x larger
- We could remove the output spill registers adding one cycle of latency (removing another 10% of the area of FP32 Tiny-FPU) → **~1.4x larger**

Preliminary results - performance

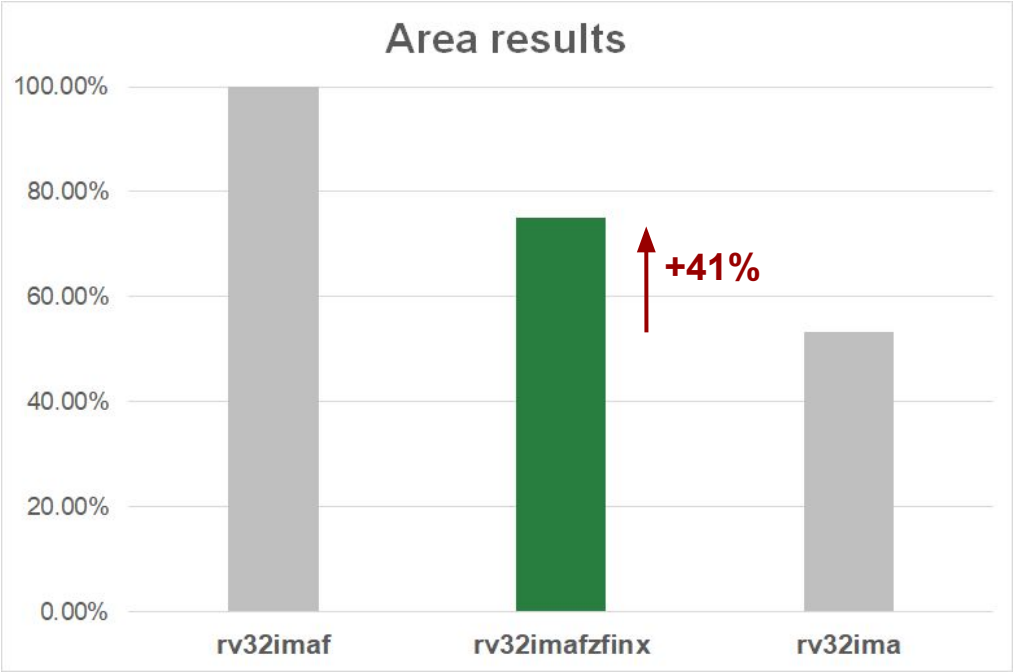
- Modified the Tiny-FPU adding one FSM stage (end of computation)
- Tiny-FPU's outputs are outputs of registers
- The input were already connected directly to the internal registers



Area Results



Area Results - no FPU spill registers



Tiny-FPU Zfinx

NEXT STEPS:

- Run synthetic and real programs on the Zfinx version
- Performance comparison