

Maxeler project

Euclidean algorithm for computing
the greatest common divisor

Dr. Veljko M. Milutinović

Vjeran Rađa
France Petač

FRI 2013/14

GCD on control flow

- Easy to implement, many different implementations
- Time complexity dependent on number of digits

```
function gcd(a, b)
  while b ≠ 0
    t := b
    b := a mod b
    a := t
  return a
```

GCD from CF to DF

- No while loops
- End condition
- Mod function
- Output

```
function gcd(a, b)
  while b ≠ 0
    t := b
    b := a mod b
    a := t
  return a
```

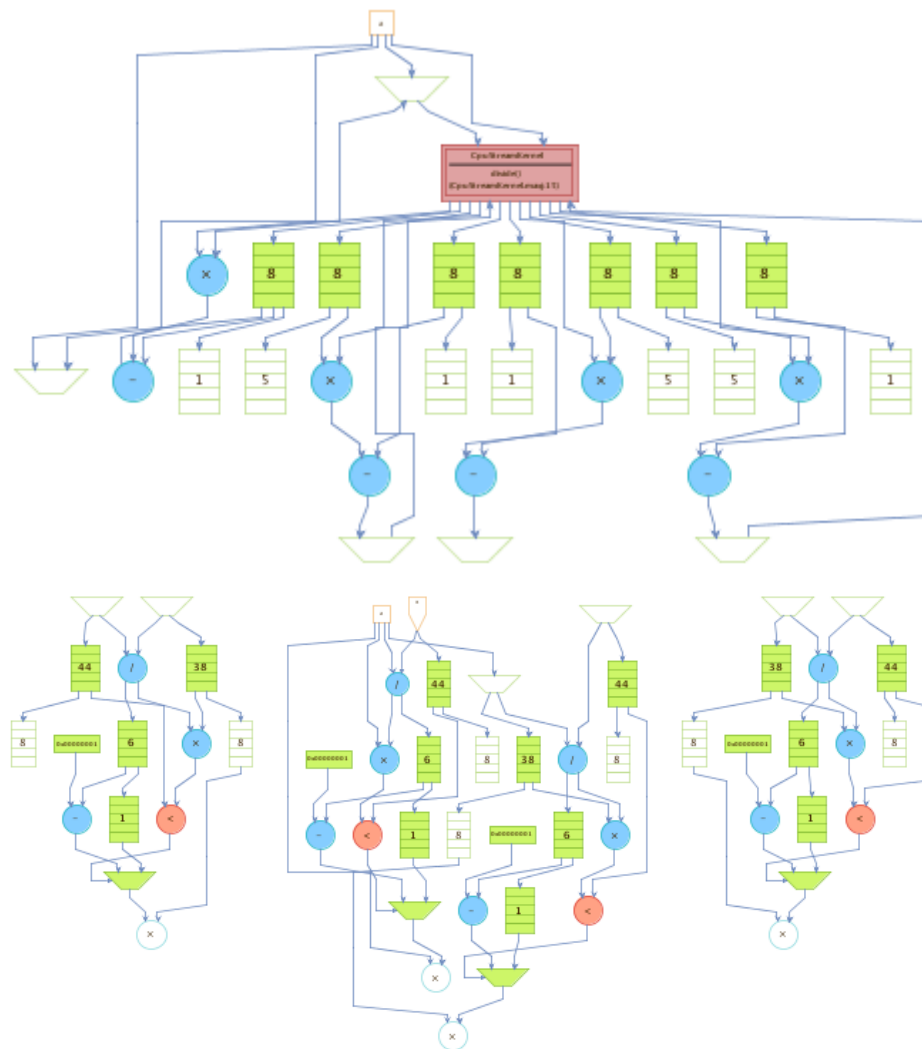
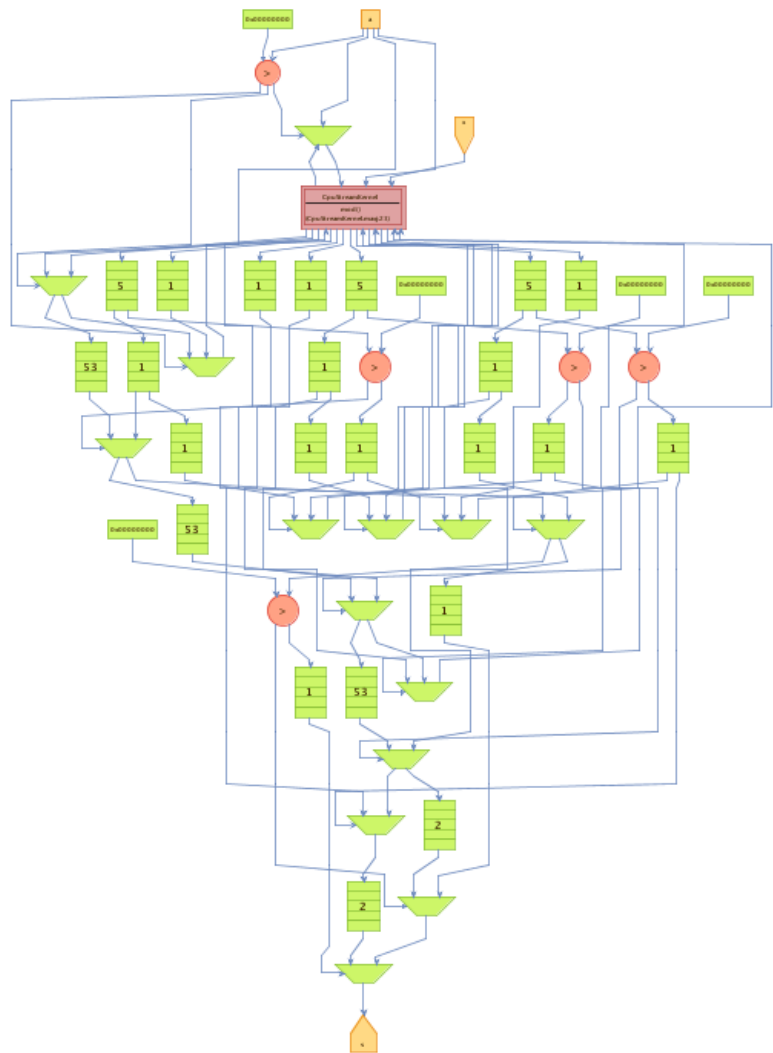
Implementation

- While to for
- End condition
- Mod function
- Division correction
- Output

```
function gcd(a, b)
  while b ≠ 0
    t := b
    b := a mod b
    a := t
  return a
```

```
for (int i=5; i > 0 ; i--)
{
  DFEVar condition = constant > 0;
  t = condition ? constant : t;
  constant = condition ? mod(a, constant) : constant;
  a = condition ? t : a;
}
```

Implementation



Possible improvements

- Parallelization
- Optimization (less ifs)

- Different algorithm

Results

- Simulation
- DFE
- GOTO: possible improvements

Conclusion

- It's hard (at first)
- It's hard (even later)

- Paradigm changes