Embedded Systems Design: A Unified Hardware/Software Introduction

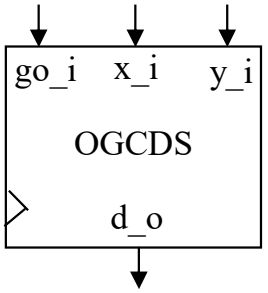
Custom single-purpose processors

Exercises

Exercise SPP_1: optimized GCD (simplified)

- Create the FSMD and the datapath for an SPP implementing the functionality described in the algorithm reported in (b)

(a) black-box view



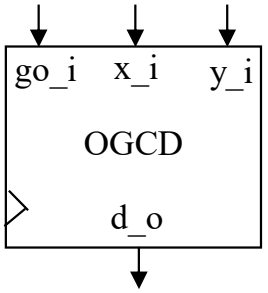
(b) desired functionality

```
// Hypothesis: x > y
0: int  x, y, r;
1: while (1) {
2:    while (!go_i);
3:    while  (y != 0)  {
4:       r = x % y;
5:       x = y;
6:       y = r;
       }
7:    d_o = x;
     }
```

Exercise SPP_2: optimized GCD

- Create the FSMD and the datapath for an SPP implementing the functionality described in the algorithm reported in (b)

(a) black-box view

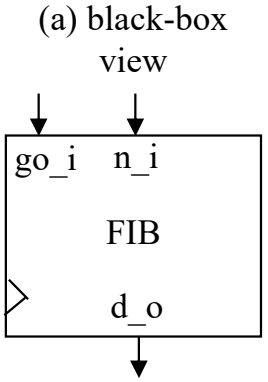


(b) desired functionality

```
0: int  x, y, r;
1: while (1) {
2:    while (!go_i);
      // x must be the larger number
3:    if (x_i >= y_i)  {
4:       x=x_i;
5:       y=y_i;
      }
6:    else {
7:       x=y_i;
8:       y=x_i;
      }
9:    while  (y != 0)  {
10:      r = x % y;
11:      x = y;
12:      y = r;
      }
13:   d_o = x;
   }
```

Exercise SPP_3: Fibonacci

- Write the algorithm and create the corresponding FSMD and datapath for an SPP, which structure is reported in (a), able to evaluate the *n_i-th* element of the Fibonacci sequence



Exercise SPP_4: Factorial

- Write the algorithm and create the corresponding FSMD and datapath for an SPP, which structure is reported in (a), able to evaluate the factorial of *n_i*

