Arduino for FPGA

This page illustrates the implementation of « sketches » on FPGA with the psC language.

Code for Arduino

The code for Arduino includes the initialization code and a main loop. Counter example:

```
enum Cmd_t
    { cReset, cUp, cDown, cLoad };

void setup()
{
    Value = 0;
}

void loop()
{
    switch(iCmd)
    {
        case cReset:
            Value = 0b;
            break();
        case cUp:
            Value++;
            break();
        case cDown:
            Value--;
            break();
        case cLoad:
            Value = NewValue;
    };
};  // Infinite loop
```

Code for Arduino FPGA

The psC language is a parallel language, but the keyword “sequential” is used to create a sequential function. Counter example:

```
enum Cmd_t
    { cReset, cUp, cDown, cLoad };

start()
{
    Value = 0;
}

sequential Counter()
{
    switch(iCmd)
    {
        case cReset:
            Value = 0b;
            break();
        case cUp:
            Value++;
            break();
        case cDown:
            Value--;
            break();
        case cLoad:
            Value = NewValue;
    };
};  // Infinite loop
```

Arduino loop

The loop code is running on a single processor executing many tasks. For example:

```
loop()
{
  Task1
  Task2
  Task3
  Task4
  Task5
}
```

Arduino FPGA loops

With the psC language, tasks in the loop on the left are executed on separate processors running in parallel.

Processors communicates with each other using signal, carrying values and events.