

# GPStudio manual v1.21

Sebastien CAUX

March 3, 2017

# Contents

	Page
<b>1 Concepts</b>	<b>4</b>
1.1 Blocks	4
1.2 Flows	4
1.3 Node	4
<b>2 Block</b>	<b>5</b>
2.1 Structure	5
2.1.1 flow interfaces	6
2.1.2 interface parameter interconnect	6
2.1.3 internal registers list	6
2.1.4 clocks list	6
2.1.5 resets list	6
2.2 Different types of block	6
2.2.1 Process	6
2.2.2 IO	6
<b>3 Flows</b>	<b>7</b>
3.1 Assumptions	7
3.2 Structure	7
3.3 Timing	8
3.4 Clock	8
3.5 Flow properties	8
<b>4 Camera Node</b>	<b>9</b>
4.1 Top level of a node	9
4.1.1 Special blocks - Network on chip	9
4.2 Properties	11
<b>5 Distribution</b>	<b>12</b>
5.1 Files organisation	12
5.2 Files format	12
5.2.1 .dev	12
5.2.2 .proc	12
5.2.3 .io	13
5.2.4 .node	13
5.3 Tools	13
5.3.1 gpnode	13
5.3.2 gplib	13
5.3.3 gpproc	13
5.3.4 gpdevice	13

5.3.5	gpviewer	13
5.4	Work flow	13

# Introduction

Smart Cameras allow to embed image processing nearest to image sensors. Embedded systems are therefore deployed to extract high level image features with (pixel-wise) computational intensive operations. Given the limited computational capabilities of embedded devices, programmable FPGA architectures have been proposed to speed-up the processing performance. Such architectures are able to perform high throughput processing within a modular and flexible approach. However, in developing FPGA vision applications one of the biggest issues is the increased development time compared to classical computer vision techniques.

Therefore, with the integration of FPGA devices into smart camera platforms, node programmability becomes a major concern for FPGA developers. In addition of the algorithm transcription using low-level programming languages such as VHDL or Verilog, FPGA designers have also to manage internal circuitry. With the time-to-market era, reducing the development time can be a considerable advantage. In this context, a higher level description would be useful to abstract low-level hardware considerations and increase the added value.

In this work we propose the *GPStudio* toolchain. It aims at managing low-level hardware abstractions allowing developers to focus on the porting of algorithms into hardware architecture. By leveraging the modular architecture concept, available Intellectual Properties (IP) modules are easily instantiated with standard interfaces between sensors, processing and communication blocks. In this way, a cross-platform IP library is proposed to improve the code re-usability for a wide range of smart camera applications. Once the application is defined, *GPStudio* automatically generates the architecture and the glue code for the targeted board. All in all, *GPStudio* allows a rapid FPGA deployment using in-library or custom defined modules (with your preferred HDL language or HLS). With the generated *GPStudio* debug facilities, the FPGA development has never been so easy!

# Chapter 1

## Concepts

Schema node ex

### 1.1 Blocks

2 types de blocks

IO block is an interface between exterior of the node and the inside. It could be a communication with others nodes or sensors.

A process block have the capability to transform data witch come from input to processed data depending of internal parameters.

$$f_{out_i} = f(\underline{f_{in}}, \underline{parameters}) \quad (1.1)$$

It have at least one input and one output, parameters are optional.

### 1.2 Flows

Flow is a dedicated connexion between a flow interface input and a flow interface output of two blocks. dataflow KPN

### 1.3 Node

A node in GPStudio is a physical platform, it can be a smart camera or a sensor.

# Chapter 2

# Block

## 2.1 Structure

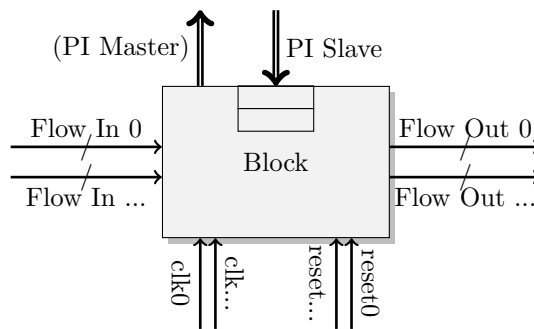


Figure 2.1: Generic block

A block is composed of :

**flow interfaces list** with direction and size of data in bits

**interface parameter interconnect** configuration of slave and master

**internal registers list** with relative address

**clocks list** with frequency, domain and shifting phase

**resets list** with polarity and reset group

**2.1.1 flow interfaces****2.1.2 interface parameter interconnect****2.1.3 internal registers list****2.1.4 clocks list****2.1.5 resets list****2.2 Different types of block****2.2.1 Process**

A process is a standard block without any additional data. It has at least one flow input and one flow output. Internal dynamic parameter with slave parameter interface (slave PI) are optional.

**2.2.2 IO**

An IO is a block with a custom interface externally connected to the node. For example, an image sensor interface is an IO because it has special connections to the outside.

# Chapter 3

## Flows

### 3.1 Assumptions

We assume that the whole design is synchronous to a single clock, `clk_proc`.

logic mux

...

### 3.2 Structure

Flow connexion is composed by three electric signals :

- a data bus of the size of one data with `_data` suffix
- a data valid signal to indicate that a data is present on the data bus with `_dv` suffix
- a flow valid signal to delimit the beginning of the flow and the end with `_fv` suffix

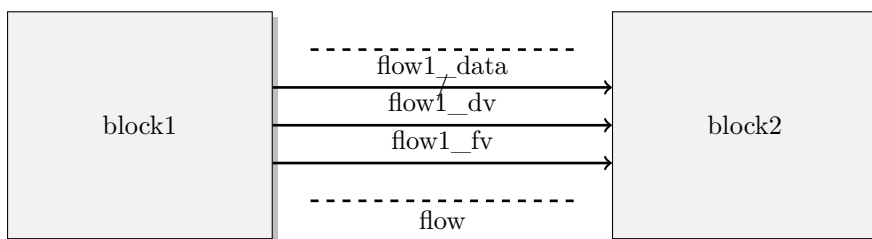


Figure 3.1: Details on flow connexion

All theses signals are synchronous on `clk_proc` clock.



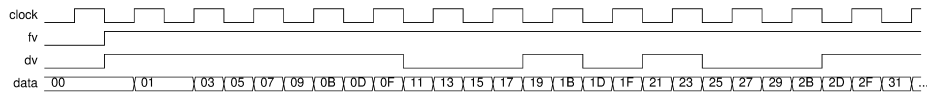


Figure 3.2: Example timing for flow signals

### 3.3 Timing

### 3.4 Clock

All the flow transmission are synchronous on the same clock, `clk_proc`. Like that, we don't need diferent clock domain for data transmission. If a process or IO need to internally work with a different clock, clock synchronisation have to be done internally of the block.

### 3.5 Flow properties

# Chapter 4

## Camera Node

### 4.1 Top level of a node

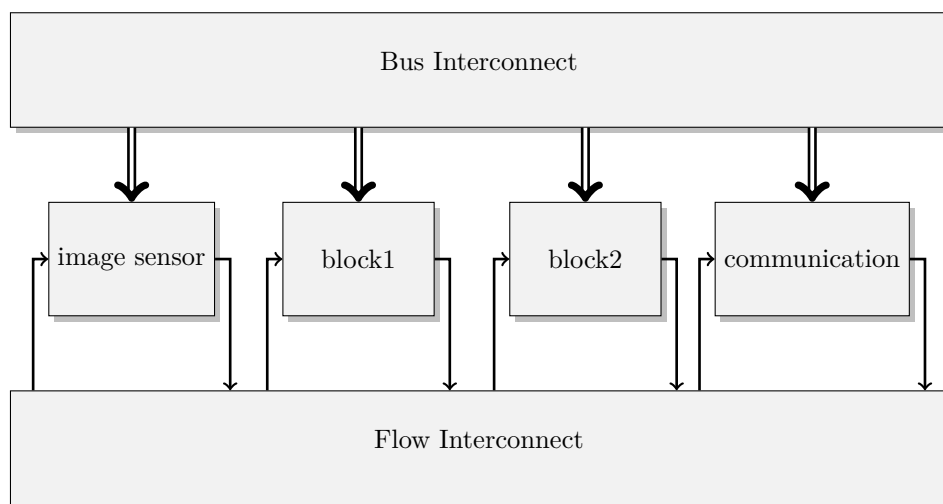


Figure 4.1: Top level of a node

#### 4.1.1 Special blocks - Network on chip

intro noc is a special block generated by GPS  
schema avec FI/PI/CI

##### **FI - Flow Interconnect**

FI - Flow Interconnect connects and re-assign mapping between flow interface blocks.

All the flow interface of each blocks are connected to FI.

This mapping rules are apply :

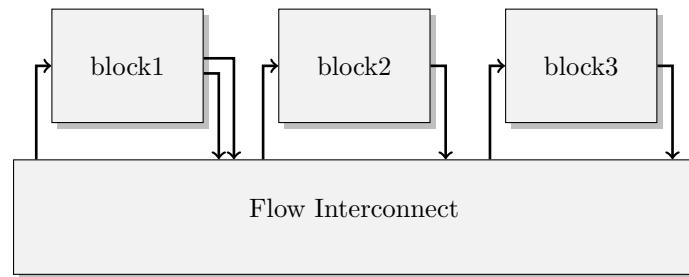


Figure 4.2: FI connexions to blocks

- In case of a direct connexion of a flow out interface to one flow input, interface are hard connected through FI.
- If two flow out are connected to one input, FI create a dynamically configurable multiplexer to permit flow redirection at runtime. This multiplexer is controlled by an internal register.

For the size of flow, FI generator process like that :

- In case of identical size of flow, data of flow interface are connected bit to bit.
- If the output is larger than the input, an msb connexion is done by default unless you explicitly specify an lsb connexion. Other bits are losted.
- And if the output is smaller than the input, an msb connexion is done by default unless you explicitly specify an lsb connexion. Zero bits are added as extra bits.

ZOOM + explication mux

This block need a special configuration specified in the node project file. This configuration contains all flow connexions between block. It's automatically added by an HDL Toolchain.

### PI - Parameter Interconnect

PI - Parameter Interconnect is a special block generated by GPS to read/write dynamic parameters of blocks. It connect all masters and slaves BI interface together and create logic control for selecting address space of blocks.

BUS

This block doesn't need any configuration. It's automatically added by an HDL Toolchain.

### CI - Clock Interconnect

CI or Clock Interconnect is a special block generated by GPS to manage all clocks in the node. It distribute the requested clocks frequency to others blocks taking care of clock domains and clock shifting.

It contains PLL if you need frequency that board can't provide.

This block need a special configuration specified in the node project file. This configuration contains frequency for-each clock domain. It's automatically added by an HDL Toolchain.

## 4.2 Properties

Properties is a way to link block low level hardware registers to high level software controller.

# Chapter 5

## Distribution

### 5.1 Files organisation

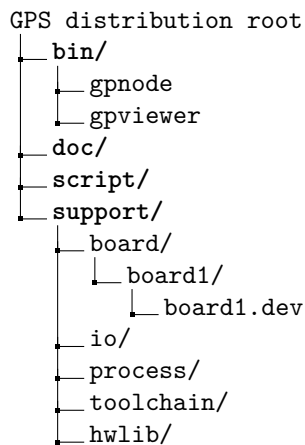


Figure 5.1: Files tree of the main distribution of GPStudio

### 5.2 Files format

All files used by GPStudio are under XML format with specific extension.

#### 5.2.1 .dev

This file is the definition of a board platform, his capabilities and functionalities.

It contains the toolchain to produce a byte code, special attributes for supporting the board and the list of available peripheral. For each peripheral,

#### 5.2.2 .proc

files implementation list with type of file and path

### 5.2.3 .io

Io file definition is quite similar of process file with addition of an external interface.

### 5.2.4 .node

## 5.3 Tools

### 5.3.1 gpnode

gpnode is a command line tool that permit to manage a gptudio node project.

*See also gpnode manual.*

### 5.3.2 gplib

### 5.3.3 gpproc

### 5.3.4 gpdevice

### 5.3.5 gpviewer

gpviewer is a graphical interface to see results of processing and setting up your smart camera during execution.

*See also gpviewer manual.*

## 5.4 Work flow