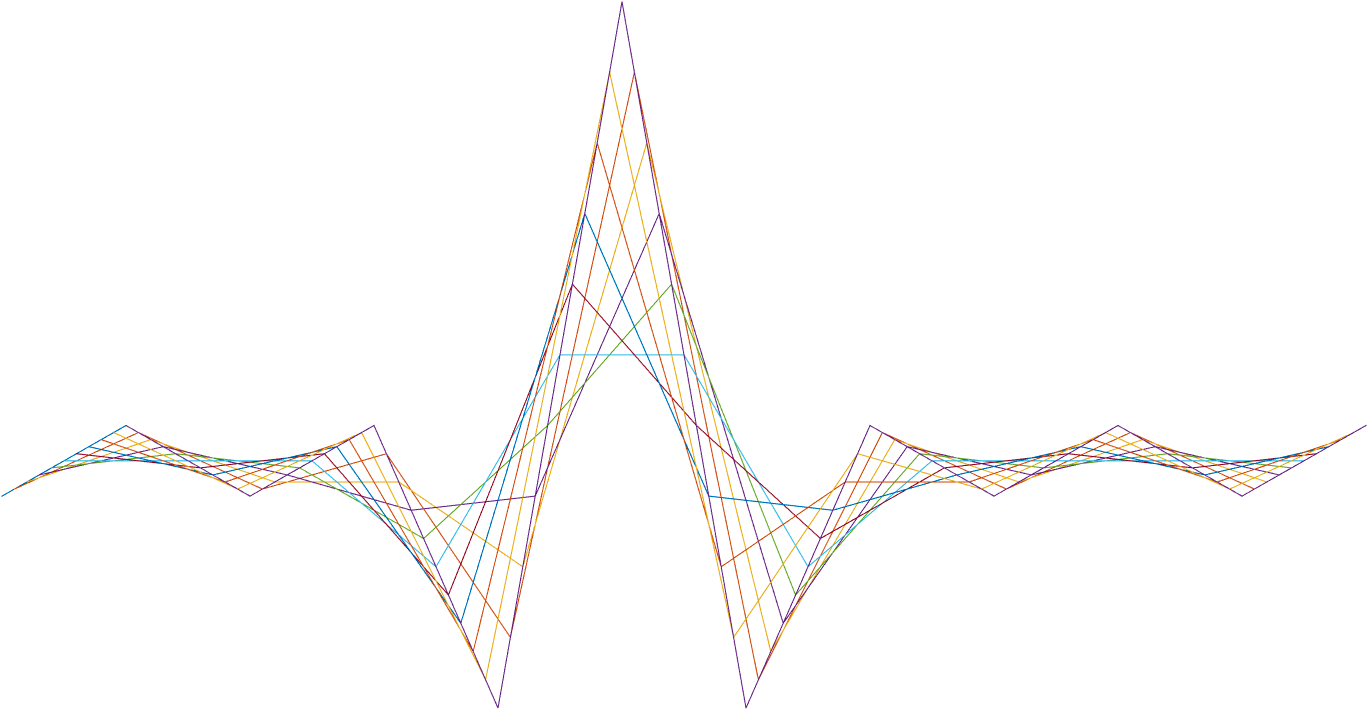
IVCAM2.0 3D Imaging Camera



ASIC A0 Packer specification

22 December 2016

Revision 0.8.0

Intel Top Secret

Table 1: Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Matlab Version | Revision Number | Revised by | Description | Revision Date |
|  |  | Ohad Menashe | Initial Release | 22 Dec 2016 |
|  |  | Yoni Chechik | redesign | 05 Jan 2017 |

7

Contents

[1 Introduction 3](#_Toc471392314)

[2 Interfaces and memory 4](#_Toc471392315)

[2.1 Interfaces 4](#_Toc471392316)

[2.1.1 Input 4](#_Toc471392317)

[2.1.2 Output 4](#_Toc471392318)

[2.2 Memory 5](#_Toc471392319)

[3 Detailed description 6](#_Toc471392320)

[3.1 Packing 6](#_Toc471392321)

[3.2 Privacy mode 6](#_Toc471392322)

[4 Registers 7](#_Toc471392323)

[5 Test plan 8](#_Toc471392324)

Introduction

The packer block is the last block in the pipe. It’s has dual purpose:

1. Pack the data and prepare it for the USB/mipi controller
2. Set privacy mode

Interfaces and memory

Interfaces

Input

* depth: 16-bit of processed depth data (ROI).
* ir: 8-bit of processed IR data (ROI).
* confidence: 4-bit of processed confidence data (ROI).

Output

* 3 stream channels of 64b each the output different configurations of the data.

Memory

N/A

Detailed description

Packing

As the data being processed in the pipe can be of any resolution, padding should be added to satisfy the controller interfaces. As calculated in The DIGG, the underlying resolution, defined by the firmware is truncated by the margin, as the pipe actual resolution differs from it. The packer padded the frame (from both the left, right, top, and bottom) to the original frame size. The padding width/length are defined by registers regsPCKRxPad & resPCKRyPad (which are auto generated). To maintain the original pipe constraints regarding invalid pixels, the padded values are fixed to 0 in stream1, 2 & 3.

In the non- padded area of each stream we will output several different modes of output that is determent by registers:

if(regs.PCKR.selFg)

%take stuff from frame grabber

else

if(regs.PCKR.allInDepth)

stream1=|Cn+1 1'h |1'h0 |In+1 2'h |Zn+1 4'h|Cn 1'h |1'h0 |In 2'h |Zn 4'h|;

else

stream1 = [];

end

if(regs.PCKR.depthEn)

stream1 = | Dn+3 4'h|...|Dn+0 4'h|

else

stream1 = [];

end

if(regs.PCKR.confEn)

stream2 = | Cn+15 1'h|...|Cn+0 1'h|

else

stream2 = [];

end

if(regs.PCKR.irEn)

stream3 = | In+7 2'h|...|In+0 2'h|

else

stream3 = [];

end

end

end

Privacy mode

When enabled, privacy mode ignores the input pixels and replaces it with a preset value: for each input – d/ir/c- we output the value at PCKRprivacyZ || PCKRprivacyC || PCKRprivacyI accordingly, and also pad with zeros like in “Packing”.

Registers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Size** | **Default** | **Range** | **Special values/ description** |
| **PCKR** |  |  |  |  |
| PCKRbypass | 1 | 0 | [0:1] |  |
| PCKRprivacyEn | 1 | 0 | [0:1] |  |
| PCKRprivacyZ | 16 | 0 | [0:2^16-1] |  |
| PCKRprivacyI | 8 | 0 | [0:2^8-1] |  |
| PCKRprivacyC | 8 | 0 | [0:2^8-1] |  |
| PCKRallInDepth | 1 | 0 | [0:1] |  |
| PCKRdepthEn | 1 | 1 | [0:1] |  |
| PCKRconfEn | 1 | 1 | [0:1] |  |
| PCKRirEn | 1 | 1 | [0:1] |  |

**Table 7: Registers**

Test plan

|  |  |  |
| --- | --- | --- |
| **Name** | **value** | **distribution** |
| **General** |  |  |
|  |  |  |
| **PCKR** |  |  |
|  |  |  |

**Table 8: Test plan**