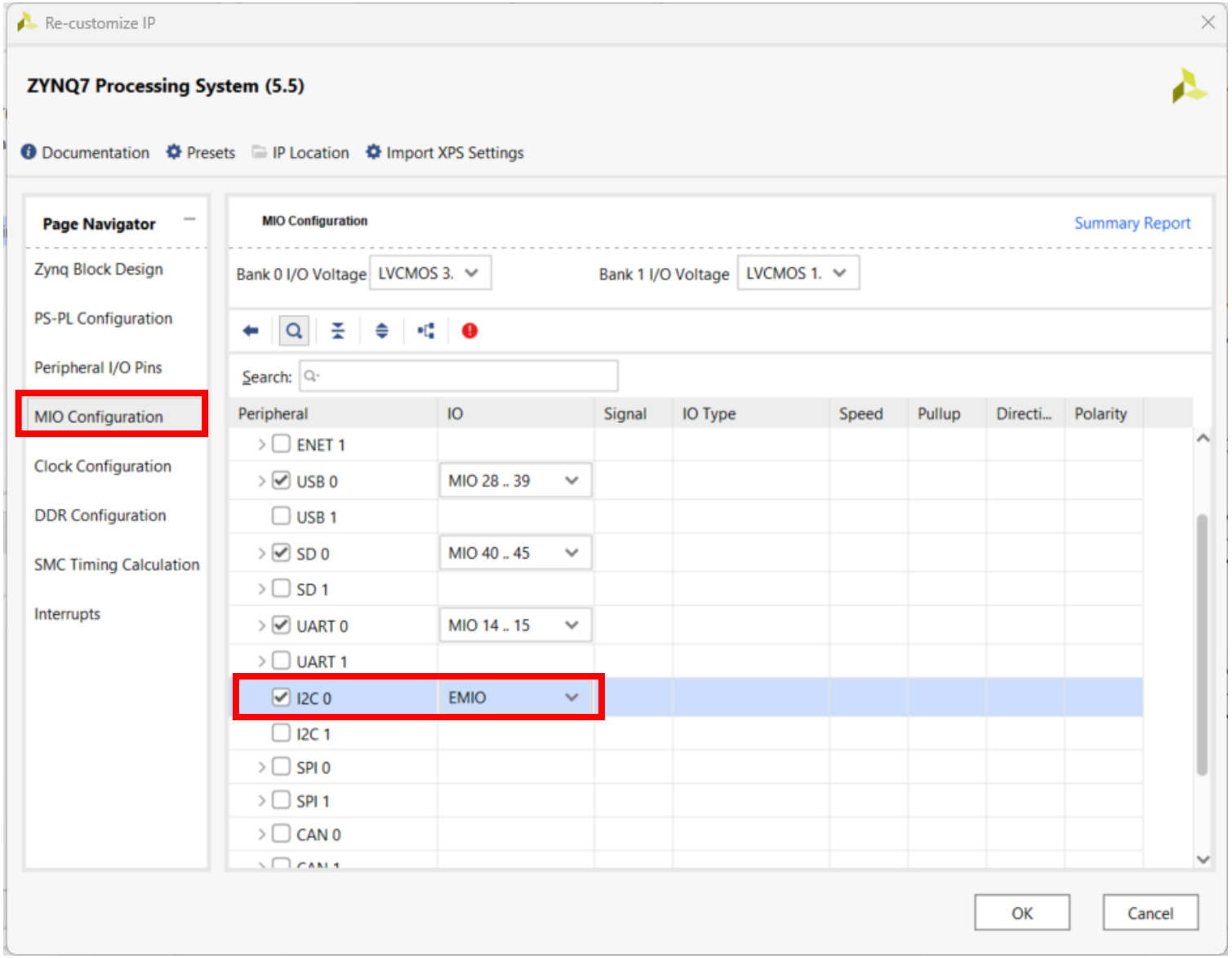
Projektiranje programabilnih SoC platformi

Predavanja FER, 2022.

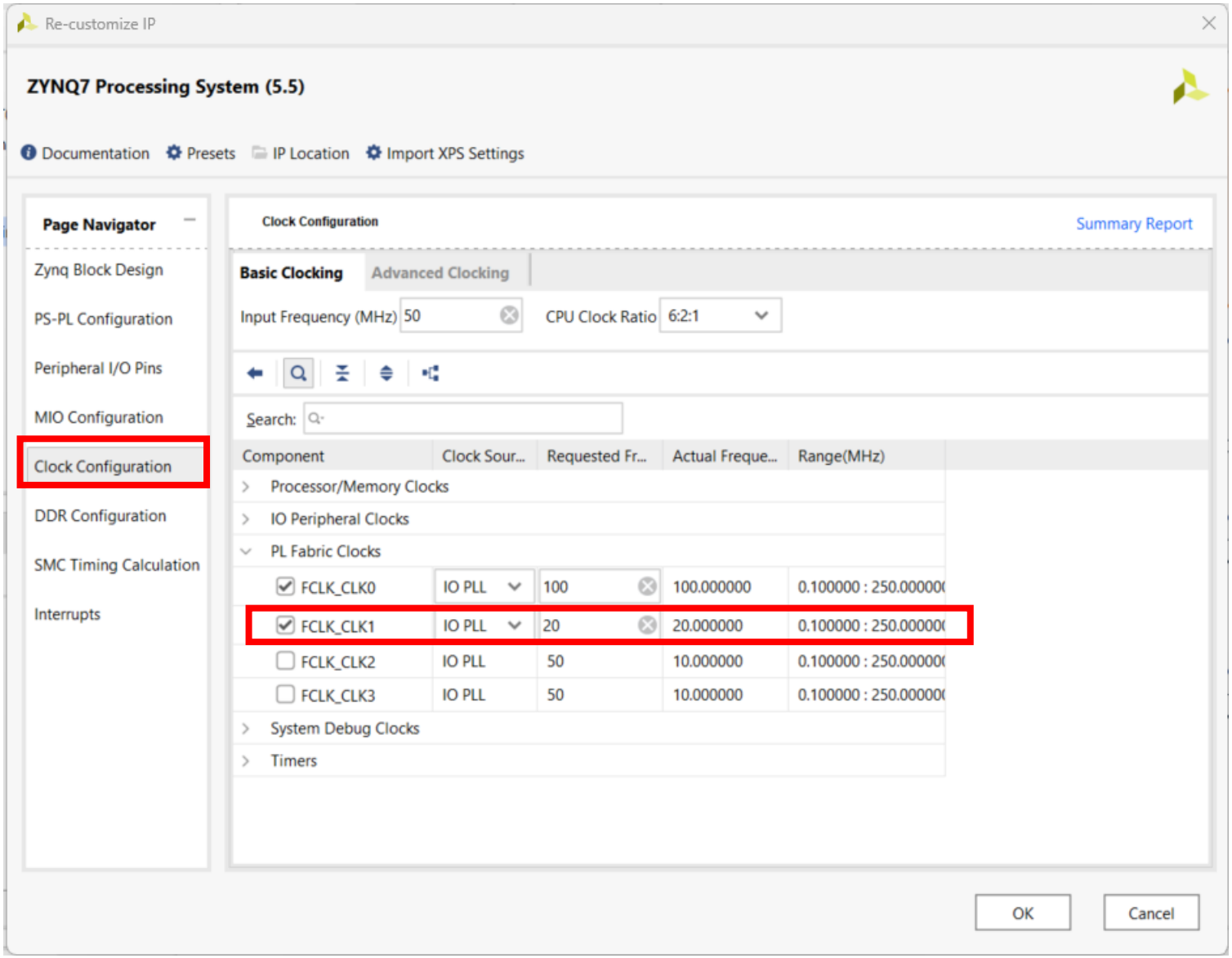
Dodavanje podrške za I2C

- Potrebno je modificirati postavke ZYNQ7 procesora
  - Proslijediti I2C priključke iz PS u PL i spojiti sa kamerom
    - Moguće je koristiti AXI IIC IP i dodati u dizajn, ali kako procesor već ima dva I2C sučelja onda to nema smisla
  - Da bi kamera radila moramo joj dovesti takt od 20MHz
    - Procesor ima mogućnost generirati četiri različita takta. Jedan se već koristi za upravljanje sa AXI komponentama (FCLK_CLK0) i postavljen je na 100MHz

Dodavanje podrške za I2C



Dodavanje takta od 20MHz



Postavljanje priključaka

- Nakon konfiguracije I2C i takta na komponenti ZYNQ7 pojavljuju se dva nova signala:
  - IIC_0
  - FCLK_CLK1

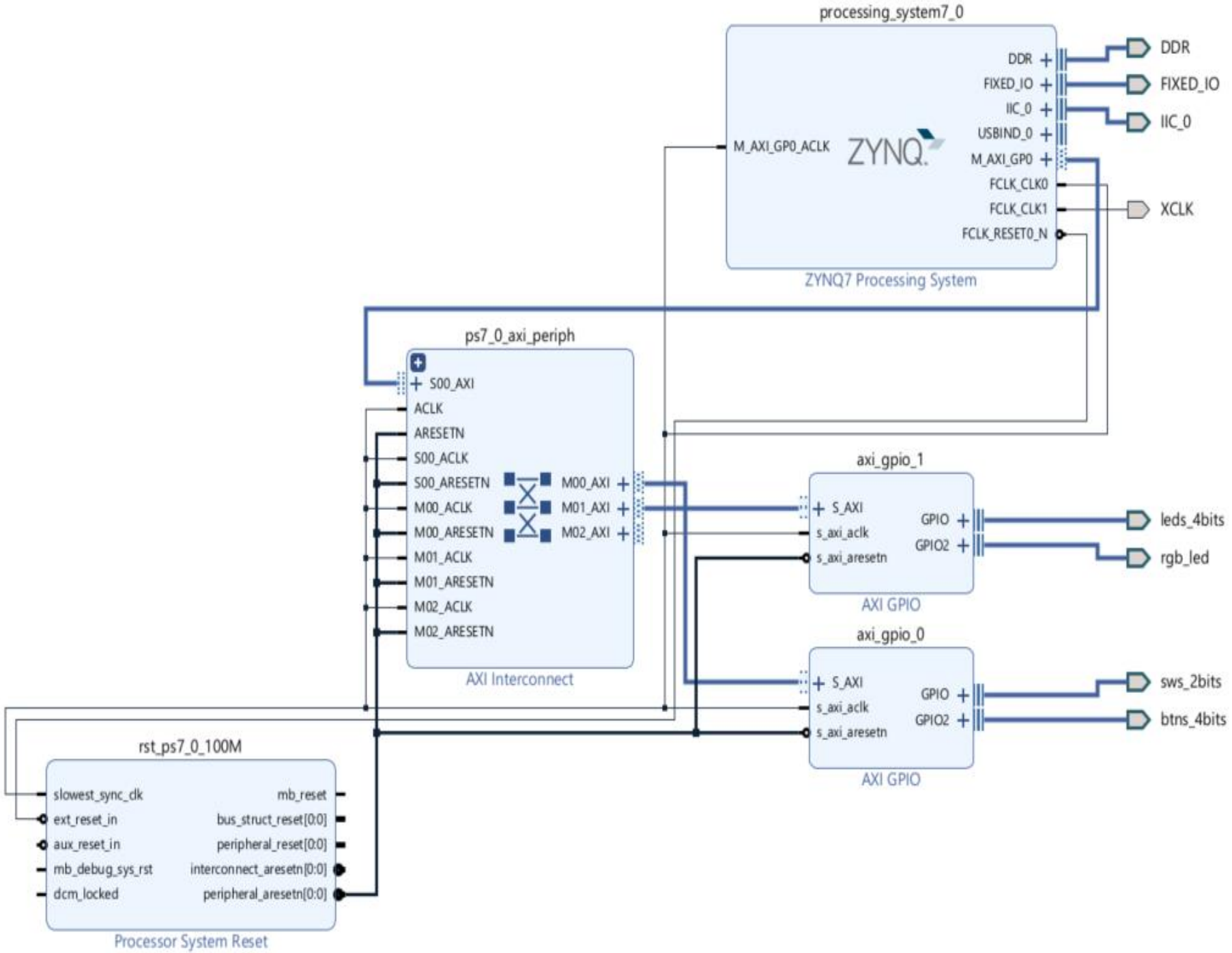
- Desnim klikom na priključak pojavljuje se meni u kojem treba odabrati „Make External”
- U prozoru Property („External Port Property”) možete promijeniti naziv priključka

- Potrebno je napraviti dvostruki klik na exportani priključak FCLK_CLK1 i postaviti da je frekvencija 20MHz

- Da bi Xilinx alati znali na koje fizičke priključke spojiti naše exportane priključke potrebno je dodati Constraints u kojem je definirano povezivanje:

```
set_property PACKAGE_PIN Y19 [get_ports XCLK]
set_property IOSTANDARD LVCMOS33 [get_ports XCLK]
set_property SLEW FAST [get_ports XCLK]
set_property OFFCHIP_TERM NONE [get_ports XCLK]

set_property PACKAGE_PIN T10 [get_ports IIC_0_scl_io]
set_property PACKAGE_PIN Y18 [get_ports IIC_0_sda_io]
set_property PULLUP true [get_ports IIC_0_sda_io]
set_property PULLUP true [get_ports IIC_0_scl_io]
set_property IOSTANDARD LVCMOS33 [get_ports IIC_0_scl_io]
set_property IOSTANDARD LVCMOS33 [get_ports IIC_0_sda_io]
```


processing_system7_0
DDR
FIXED_IO
IIC_0
USBIND_0
M_AXI_GP0
FCLK_CLK0
FCLK_CLK1
FCLK_RESET0_N
M_AXI_GP0_ACLK
ZYNQ
ZYNQ7 Processing System
DDR
FIXED_IO
IIC_0
XCLK
ps7_0_axi_periph
S00_AXI
ACLK
ARESETN
S00_ACLK
S00_ARESETN
M00_ACLK
M00_ARESETN
M01_ACLK
M01_ARESETN
M02_ACLK
M02_ARESETN
M00_AXI
M01_AXI
M02_AXI
AXI Interconnect
axi_gpio_1
S_AXI
s_axi_aclk
s_axi_aresetn
GPIO
GPIO2
AXI GPIO
leds_4bits
rgb_led
axi_gpio_0
S_AXI
s_axi_aclk
s_axi_aresetn
GPIO
GPIO2
AXI GPIO
sws_2bits
btns_4bits
rst_ps7_0_100M
slowest_sync_clk
ext_reset_in
aux_reset_in
mb_debug_sys_rst
dcm_locked
mb_reset
bus_struct_reset[0:0]
peripheral_reset[0:0]
interconnect_aresetn[0:0]
peripheral_aresetn[0:0]
Processor System Reset

iicps_v3_15

- The „xiicps.h” header file is an implementation of IIC driver in the PS block.

- Za izvedbu i korištenje proučiti:
  - xiicps_polled_master_example

- Inicializacija:

```
/*
 * Initialize the IIC driver so that it's ready to use
 * Look up the configuration in the config table,
 * then initialize it.
 */
Config = XIicPs_LookupConfig(DeviceId);
if (NULL == Config) {
    return XST_FAILURE;
}

Status = XIicPs_CfgInitialize(&Iic, Config, Config->BaseAddress);
if (Status != XST_SUCCESS) {
    return XST_FAILURE;
}
```

iicps_v3_15


```
#define IIC_SLAVE_ADDR      0x21
#define IIC_SCLK_RATE       400000
```

- Pisanje:

```
Status = XIicPs_MasterSendPolled(&Iic, Buffer, 2, IIC_SLAVE_ADDR);
if (Status != XST_SUCCESS) return XST_FAILURE;
while (XIicPs_BusIsBusy(&Iic)) {/* NOP */}
```

- Čitanje

```
Status = XIicPs_MasterRecvPolled(&Iic, Buffer, 1, IIC_SLAVE_ADDR);
if (Status != XST_SUCCESS) return XST_FAILURE;
while (XIicPs_BusIsBusy(&Iic)) {/* NOP */}
```

Konfiguracija kamere

- Konfiguraciju kamera treba napraviti prema podatcima iz „camera_config.txt" koji se nalazi u repozitoriju predmeta

```
#define   REG_HAECC5    0xa8    /* Hist AEC/AGC control 5 */
#define   REG_HAECC6    0xa9    /* Hist AEC/AGC control 6 */
#define   REG_HAECC7    0xaa    /* Hist AEC/AGC control 7 */
#define   REG_BD60MAX   0xab    /* 60hz banding step limit */

struct regval_list {
    unsigned char reg_num;
    unsigned char value;
};

static struct regval_list ov7670_default_regs[] = {
    //{ REG_COM7, COM7_RESET },
/*
 * Clock scale: 3 = 15fps
 *              2 = 20fps
 *              1 = 30fps
 */
    //{ REG_CLKRC, 0x1F},   /* OV: clock scale (30 fps) */
    { REG_CLKRC, 0x01}, /* OV: clock scale (30 fps) */
    { REG_TSLB,  0x04 },    /* OV */
    { REG_COM7, 0 },    /* VGA */
    /*
     * Set the hardware window.  These values from OV don't entirely
     * make sense - hstop is less than hstart.  But they work...
     */
    { REG_HSTART, 0x13 },   { REG_HSTOP, 0x01 },
    { REG_HREF, 0xb6 }, { REG_VSTART, 0x02 },
    { REG_VSTOP, 0x7a },    { REG_VREF, 0x0a },

    { REG_COM3, 0 },    { REG_COM14, 0 },
    /* Mystery scaling numbers */
    { 0x70, 0x3a },     { 0x71, 0x35 },
    { 0x72, 0x11 },     { 0x73, 0xf0 },
    { 0xa2, 0x02 },     { REG_COM10, 0x0 },

    /* Gamma curve values */
    { 0x7a, 0x20 },     { 0x7b, 0x10 },
    { 0x7c, 0x1e },     { 0x7d, 0x35 },
    { 0x7e, 0x5a },     { 0x7f, 0x69 },
    { 0x80, 0x76 },     { 0x81, 0x80 },
    { 0x82, 0x88 },     { 0x83, 0x8f },
    { 0x84, 0x96 },     { 0x85, 0xa3 },
```

Spajanje kamere na AXI stream

- Potrebno je prilagoditi ZV-Port sučelje na AXI stream sučelje

- Za spajanje koristimo gotovo rješenje koje se može naći na internetu a autori su:
  - `----------------------------------------------------------`
  - `--   Authors: Mike Field <hamster@snap.net.nz>`
  - `--   Lauir Vosandi <lauri.vosandi@gmail.com>`
  - `----------------------------------------------------------`

```
entity ov7670_axi_stream_capture is
    port (
        pclk            : in  std_logic;
        vsync           : in  std_logic;
        href            : in  std_logic;
        d               : in  std_logic_vector (7 downto 0);
        m_axis_tvalid   : out std_logic;
        m_axis_tready   : in  std_logic;
        m_axis_tlast    : out std_logic;
        m_axis_tdata    : out std_logic_vector ( 31 downto 0 );
        m_axis_tuser    : out std_logic;
        aclk            : out std_logic
    );
end ov7670_axi_stream_capture;
```

- Pogledajmo rješenje