Tips for porting VGA to the CY8CKIT-059 (PSoC 5LP Stick)

Eric Ponce
Massachusetts Institute of Technology

May 13, 2019

1 Introduction

The Cypress PSoC Creator project and design files provided to `6.115` students for using VGA in their final projects were designed around a custom designed PSoC 5LP VGA board. This article introduces some tips for porting the supplied code over to the CY8CKIT-059 PSoC 5LP Evaluation Board.

2 Pin Changes

Table 4-1. J1 Header Pin Details

PSoC 5LP Prototyping Kit GPIO Header (J1)		
Pin	Signal	Description
J1_01	P2.0	GPIO
J1_02	P2.1	GPIO/LED
J1_03	P2.2	GPIO/SW
J1_04	P2.3	GPIO
J1_05	P2.4	GPIO
J1_06	P2.5	GPIO
J1_07	P2.6	GPIO
J1_08	P2.7	GPIO
J1_09	P12.7	GPIO/UART_TX
J1_10	P12.6	GPIO/UART_RX
J1_11	P12.5	GPIO
J1_12	P12.4	GPIO
J1_13	P12.3	GPIO
J1_14	P12.2	GPIO
J1_15	P12.1	GPIO/I2C_SDA
J1_16	P12.0	GPIO/I2C_SCL
J1_17	P1.0	GPIO
J1_18	P1.1	GPIO
J1_19	P1.2	GPIO
J1_20	P1.3	GPIO
J1_21	P1.4	GPIO
J1_22	P1.5	GPIO
J1_23	P1.6	GPIO
J1_24	P1.7	GPIO
J1_25	GND	Ground
J1_26	VDDIO	Power

DEBUG
EXT CIR

Table 4-2. J2 Header Pin Details

PSoC 5LP Prototyping Kit GPIO Header (J2)		
Pin	Signal	Description
J2_01	VDD	Power
J2_02	GND	Ground
J2_03	RESET	Reset
J2_04	P0.7	GPIO
J2_05	P0.6	GPIO
J2_06	P0.5	GPIO
J2_07	P0.4	GPIO/BYPASS CAP
J2_08	P0.3	GPIO/BYPASS CAP
J2_09	P0.2	GPIO/BYPASS CAP
J2_10	P0.1	GPIO
J2_11	P0.0	GPIO
J2_12	P15.5	GPIO
J2_13	P15.4	GPIO/CMOD
J2_14	P15.3	GPIO/XTAL_IN
J2_15	P15.2	GPIO/XTAL_OUT
J2_16	P15.1	GPIO
J2_17	P15.0	GPIO
J2_18	P3.7	GPIO
J2_19	P3.6	GPIO
J2_20	P3.5	GPIO
J2_21	P3.4	GPIO
J2_22	P3.3	GPIO
J2_23	P3.2	GPIO/BYPASS CAP
J2_24	P3.1	GPIO
J2_25	P3.0	GPIO
J2_26	GND	Ground

Figure 1: CY8CKIT-059 Pinout w/ Debug and Externally Loaded Pins Highlighted

The implementation requires 6 contiguous pins from a single port for the 2-bit RGB VGA output pins. On a small evaluation board with overloaded pin functionality, such as the CY8CKIT-059, this presents some constraints on the pin selection. In particular, the kit contains 7 GPIOs externally loaded by a capacitor, 2 GPIOs connected to a push button and LED, and 4 GPIOs used for debugging and programming.

The highest possible input frequency for a 1024×768 @ 60 Hz VGA display is the known as the pixel frequency and is 65 MHz. In our implementation, the pixels are grouped into 8×8 chunks. reducing the highest PSoC output frequency to 8.125 MHz. This frequency is too high to drive directly into a capacitor, so Ports 0 and 3 are not eligible for VGA output. Ports 1, 2, and 15 are also not eligible so only Port 12 is available for use as the VGA output. Even for Port 12, either choice of contiguous pin range will consume

either the UART or I2C connection to the on-board KitProg. Similar considerations will need to be taken into account for porting to other evaluation boards such as the CY8CKIT-050.

Note that PSoC Creator will not always warn you if you select improper pins, as it does not make assumptions about external circuits connected to the pins.

3 Clocks

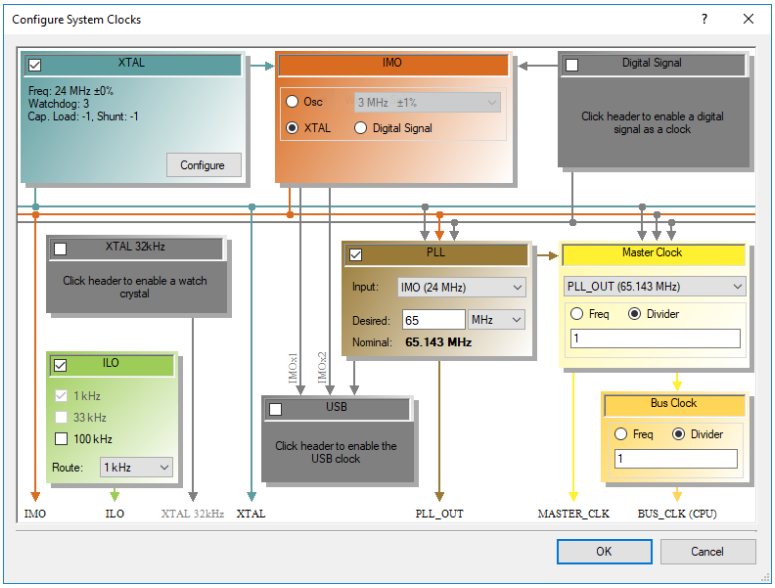

Figure 2: Default VGA Project Clock Configuration

The provided design files rely on using an external 24 MHz crystal to provide the main clock rate to the PSoC device as shown in Figure 2.

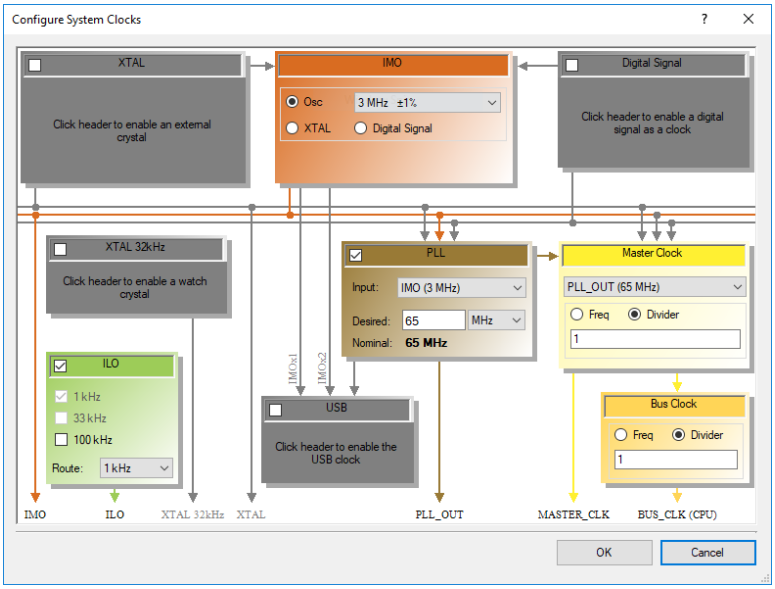

Figure 3: CY8CKIT-059 Appropriate Clock Configuration

As the CY8CKIT-059 does not include an on-board external clock, the clock configuration settings will need to be changed when porting over. A 3 MHz internal clock will suffice for most VGA implementations, and can be configured using the Creator clock configuration window as shown in Figure 3

4 Linker Script and Video Glitches

Table 5-5. Peripheral Data Address Map

Address Range	Purpose
0x00000000 – 0x0003FFFF	256 KB flash
0x1FFF8000 – 0x1FFFFFFF	32 KB SRAM in Code region
0x20000000 – 0x20007FFF	32 KB SRAM in SRAM region
0x40004000 – 0x400042FF	Clocking, PLLs, and oscillators
0x40004300 – 0x400043FF	Power management
0x40004500 – 0x400045FF	Ports interrupt control
0x40004700 – 0x400047FF	Flash programming interface
0x40004800 – 0x400048FF	Cache controller
0x40004900 – 0x400049FF	I^2C controller
0x40004E00 – 0x40004EFF	Decimator

Figure 4: PSoC CY8C58LP Family Memory Map

To improve memory performance in this VGA application, the contiguous 64 kB SRAM memory provided in the PSOC is split into two individually-accessible banks. This avoids memory port conflicts when software and DMA try to access SRAM at the same time.

Page 21 in [1] details the memory map for CY8C58LP Family PSoC devices, shown in Figure 4.

We reserve the first 32 kB SRAM Code region for normal program data and the second 32 kB SRAM block for video memory by changing the memory section at line 24 in the original linker script (`cm3gcc.ld`),

```
MEMORY
{
 rom(rx) : ORIGIN=0x0, LENGTH = 262144
 ram(rwx) : ORIGIN=0x20000000 - (65536/2), LENGTH = 65536
}
```

to what we find in our provided linker script (`custom.ld`),

```
MEMORY
{
 rom(rx) : ORIGIN=0x0, LENGTH = 262144
 ram(rwx) : ORIGIN=0x20000000 - (65536/2), LENGTH = 32768
 vram(rwx) : ORIGIN=0x20000000, LENGTH = 32768
}
```

This custom linker script is already included in the design files, as well as instructions for using it, but in the case of a non-standard PSoC with a different memory map, similar changes to the default linker script will be need to be made. In the worst case, a single SRAM block may be used, although display glitches caused by memory conflicts will be likely.

Even with a properly functioning DMA and SRAM setup, there are still multiple ways to cause glitches in the video output through normal programming of the PSoC. Memory moves throughout the PSoC on a large peripheral data bus controlled by the CPU and DMA controller (DMAC), connected by smaller-data-width spokes to individual chunks of memory as shown in Figure 5. While transfers between different spokes can occur in parallel, a single spoke can only handle one memory transaction at a given time. The DMAC in this application is constantly copying data between the SRAM and IO memory, using spokes 0 and 1 to output the pixels to the VGA connector. Thus, if your program modifications make heavy use of the IO or external memory, you are likely to cause spoke 1 conflicts between the CPU and DMAC, leading to video glitches.

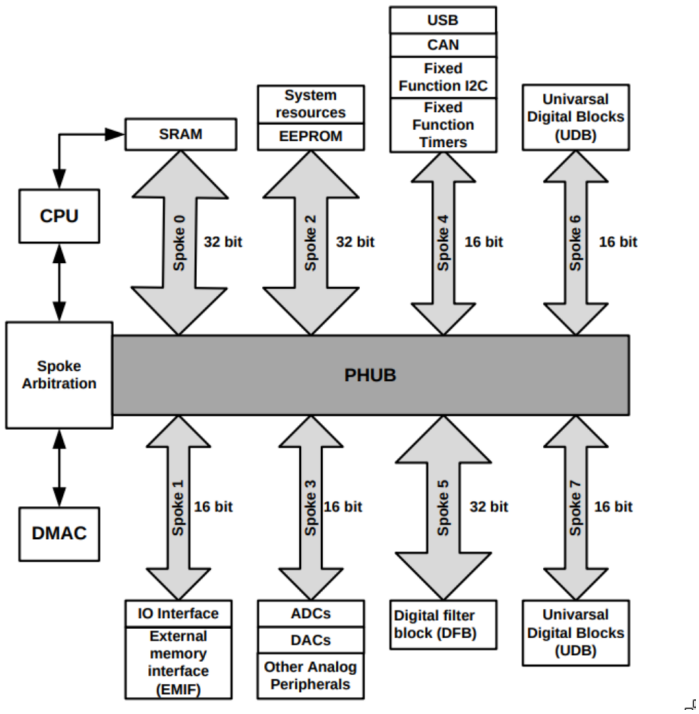

Figure 5: Internal PSoC Data Bus layout

References

[1] PSoC 5LP: CY8C58LP Family Datasheet: Programmable System-on-Chip (PSoC). [Online]. Available: https://www.cypress.com/documentation/datasheets/psoc-5lp-cy8c58lp-family-datasheet-programmable-system-chip-psoc. [Accessed: 30-Apr-2019].