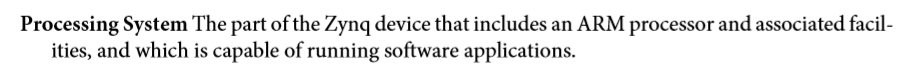
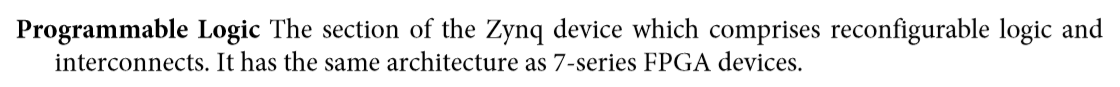
**HW2**

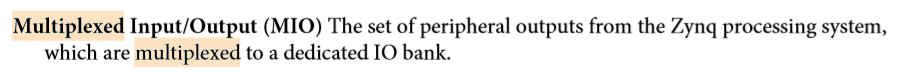
1. **Define the following terms as used in Zynq-7000 (Most definitions are copied from the Zynq Book.)**
   1. **Processing System (PS)**



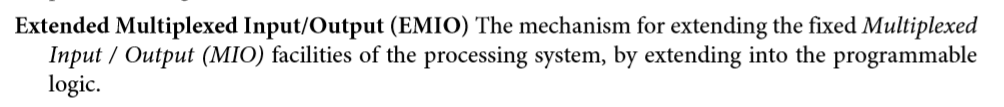
**1.2 Programmable Logic (PL)**



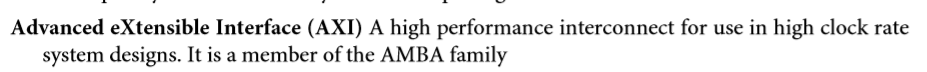
**1.3 Multiplexed IO (MIO)**



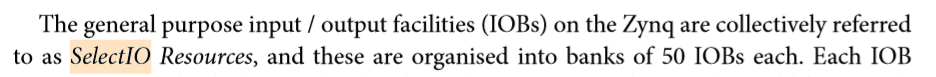
**1.4 Extended Multiplexed IO (EMIO)**



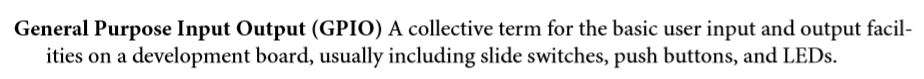
**1.5 AXI 3.0**



**1.6 SelectIO**



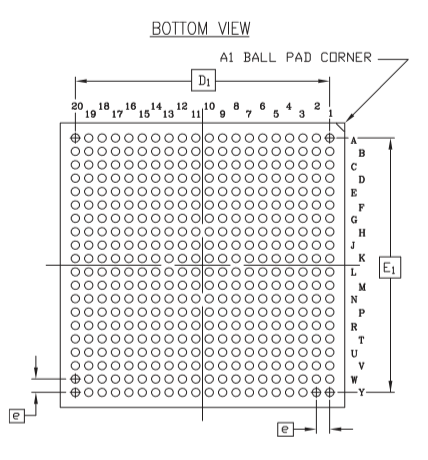
**1.7 GPIO**



1. **The Zybo’s Zynq-7010 chip bears the following marking: “XC7Z0101CLG400C”. It uses the “Ball Grid Array” (BGA) pinout for maximum packaging density. Answer the following questions about the chip. Include sources and evidences to support your answers such as document names and page numbers etc.**
   1. **What is the maximum number of physical pin connections available with this particular BGA layout? (Hint: see ug865)**

answer : 20\*20 = 400

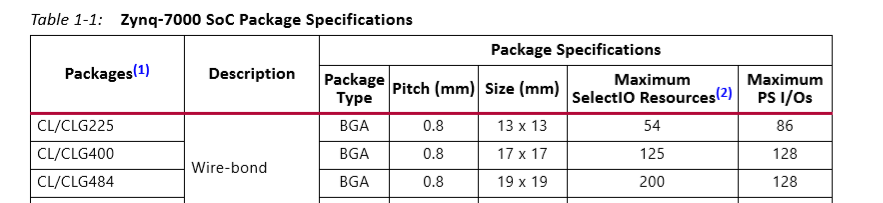
ug865-Zynq-7000-Pkg-Pinout.pdf p78



* 1. **What is the vertical and horizontal spacing (pitch) between adjacent ball grid pins?**

answer : 0.8mm

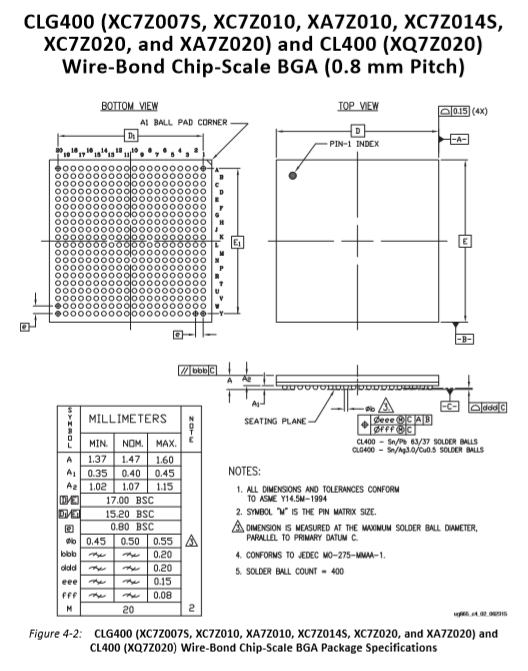
ug865-Zynq-7000-Pkg-Pinout.pdf p10



**2.3 What are the overall physical dimensions of the Zybo’s Zynq chip?**

answer : 1.60mm \* 17mm \* 17mm

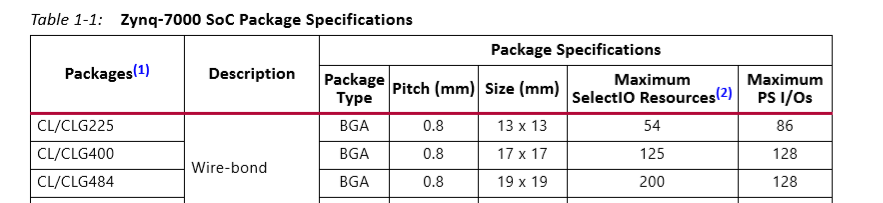
ug865-Zynq-7000-Pkg-Pinout.pdf p78



**2.4 How many processing system (PS) I/O pins are available?**

answer : 128

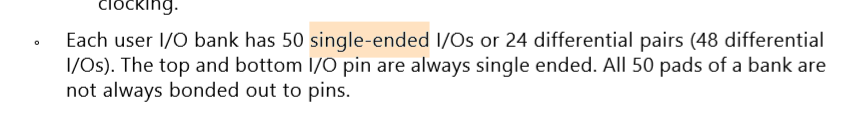
ug865-Zynq-7000-Pkg-Pinout.pdf p10



**2.5 How many programmable logic (PL) (“Select I/O”) single-ended I/O pins are available?**

answer : 50

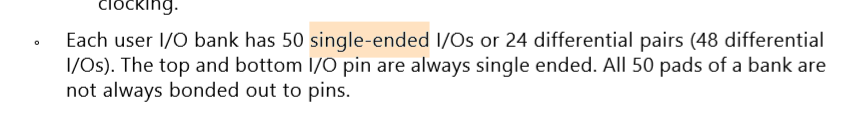
ug865-Zynq-7000-Pkg-Pinout.pdf p19



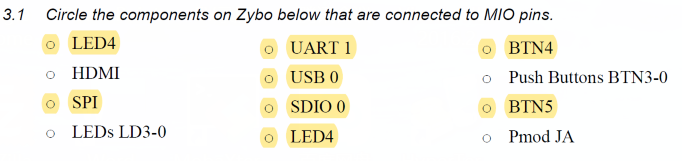
**2.6 How many programmable logic (PL) (“Select I/O”) differential I/O pairs pin pairs are available?**

answer : 24

ug865-Zynq-7000-Pkg-Pinout.pdf p19



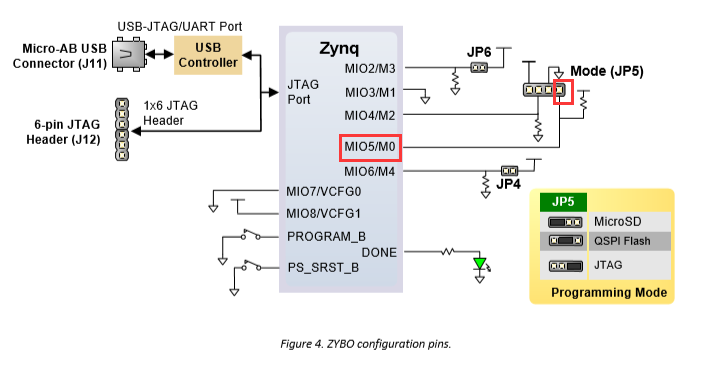
1. **Find out if I/O pins on the Zybo board are mapped to either MIO or Select I/O pins from ZYBO FPGA Board Reference Manual 2016 available from https://reference.digilentinc.com/\_media/zybo:zybo\_rm.pdf.** 
   1. **Circle the components on Zybo below that are connected to MIO pins.**



* 1. **Find where MicroSD interface pins are connected, Select I/O or MIO, and their pin numbers? You can include screen shots as your answer.**

answer : MIO5/M0

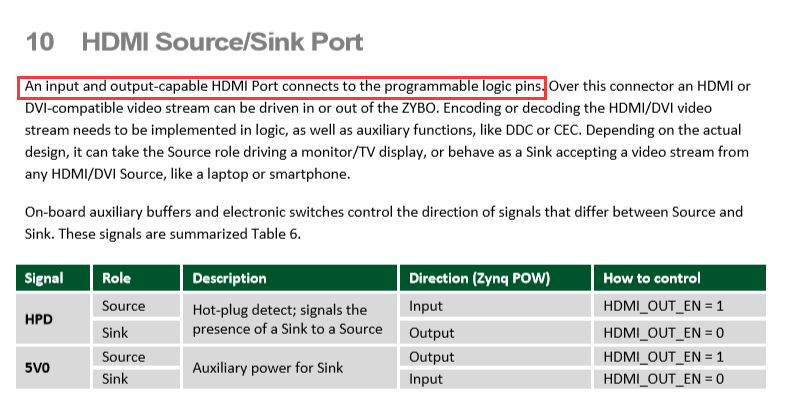
**zybo\_rm.pdf p10**



* 1. **Find where HDMI interface pins are connected, Select I/O or MIO, and their pin numbers? Include screen shots to support your answer.**

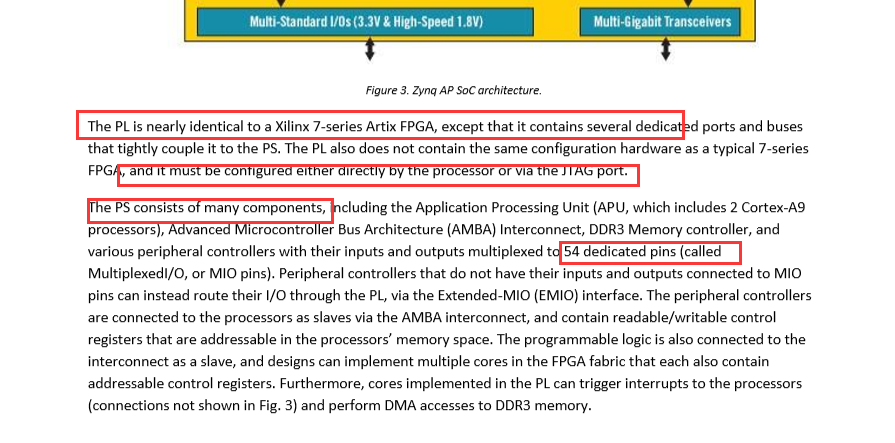
HDMI is PL pin.

zybo\_rm.pdf p16



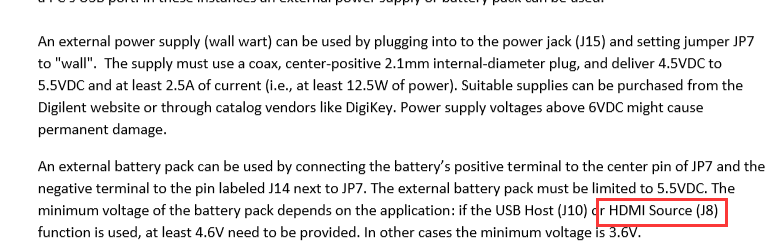
The PL has no MIO pins. Only the PS has MIO pins.

zybo\_rm.pdf p6



J8 Select I/O

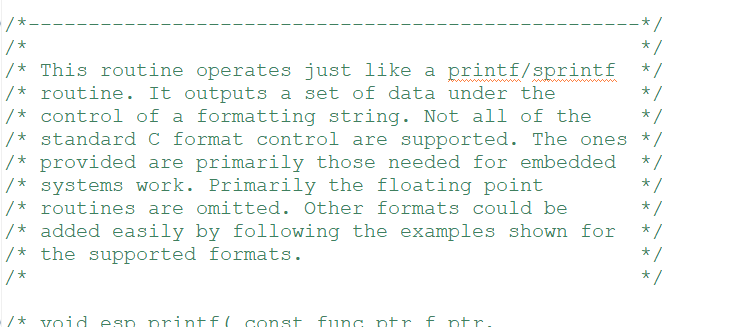
zybo\_rm.pdf p4



1. **Describe major differences between printf() and xil\_printf() and why and when xil\_printf() is better than printf(). (Hint: read the header of xil\_printf().c source code.) Include screen shots to support your conclusion.**

xil\_printf() the floating point routines are omitted.

In fact, the functions of xil\_printf and printf are the same, except that xil\_printf removes all the functions of floating point. After that, the subroutine becomes very small.

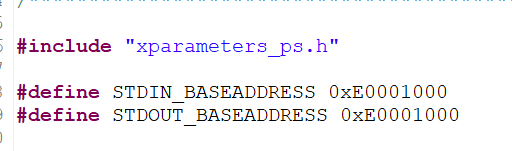


1. **Read the source code for xil\_printf(), describe what it does and how it is implemented to accomplish its function. Include screen shot or reference pages to support your conclusion.** 
   1. **Where is the base address of the STDOUT and STDIN. Describe the call structure and how the driver knows the base address of UART 1.**

STDIN\_BASEADDRESS : 0xE0001000

STDOUT\_BASEADDRESS : 0xE0001000

xparameters.h



* 1. **Describe the call structure and how**

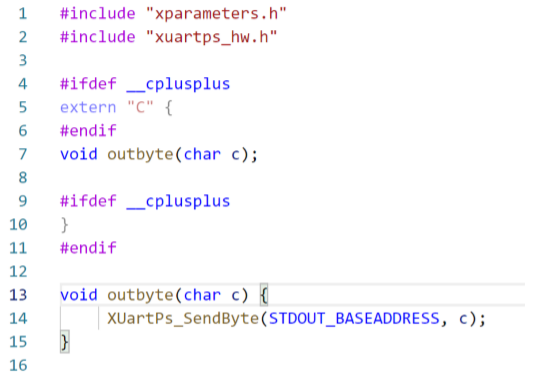
outbyte.c : 14 XuartPs\_SendByte(STDOUT\_BASEADDRESS,c);

xuartps\_hw.c:84 XuartPs\_WriteReg(BaseAddress, XUARTPS\_FIFO\_OFFSET, (u32)Data);

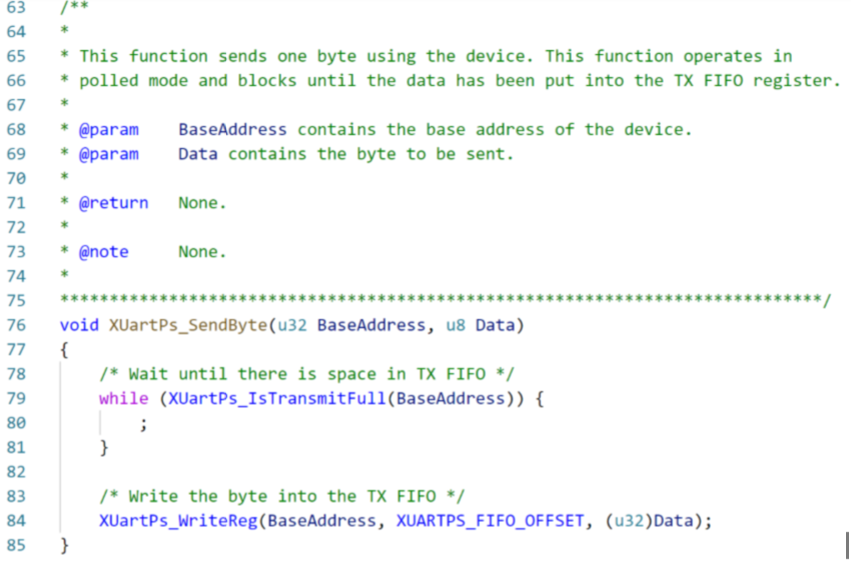
extend macro to Xil\_Out32((BaseAddress+(u32)(0x0030U),(u32)((u32)Data));

the device knows the base address from macros defined in xparameters.h and xuartps\_hw.h etc.

outbyte.c



xuartps\_hw.c



* 1. **How does xil\_printf() know where or which com port, to send its formatted character string?**

the xil\_printf() call outbyte() to write byte to com port , the outbyte() call XuartPs\_SendByte(STDOUT\_BASEADDRESS,c), where STDOUT\_BASEADDRESS is the port’s address.

* 1. **How are data received and transmitted?**

the data are write to TX FIFO, and receive from RX FIFO

* 1. **Are the sending and transmitting subroutines interrupt-driven or blocking calls?**

sending is blocking call

**5.6 Indicate how Level 1 driver of xil\_printf() relates to Level 0 driver of uart device.**

**Hint: xil\_printf() uses outbyte() from uartps\_3\_0 library, which sends data to UART 1. Include screen captures to support your answer.**

xil\_printf()

outbyte()

XuartPs\_SendByte()

XUL\_Out32(): level 1 driver

1. **Zybo Board Interface Files. Use XML Notepad from Microsoft to describe how Pmod Connector JC is defined in Board Support File A.0 for zybo-z7-10. Include screen captures to support your description. (zybo-z7-10 board files can be downloaded from**

[**https://github.com/Digilent/vivado-boards/archive/master.zip**](https://github.com/Digilent/vivado-boards/archive/master.zip)

**XML Notepad is available from**

[**http://www.lovettsoftware.com/downloads/xmlnotepad/readme.htm**](http://www.lovettsoftware.com/downloads/xmlnotepad/readme.htm)

* 1. **How is the component defined?**

The component is named jc

* 1. **How is connection defined?**

The connection is defined as part0\_jc. The two components are part0 and jc.

* 1. **Where in the board file and how is the interface defined?**

How are interfaces defined? All interfaces are defined under the first component of part0, the chip definition.

* 1. **How are the pins described and mapped between logical and physical pins?**

The pins are described and mapped in part0\_pins.xml file between logical and physical pins. However, pins 55 to 62 are not included in zybo-z7-10 A.0 part0\_pins.xml file. They are given in zybo-z7-20 A.0 part0\_pins.xml file as follows. JC1 to JC10 are connected to pins 55 to 62.each one is connected to a physical pin such as JC1 on V15, JC2 on W15, etc.