List of SOPC components

- **NIOS II processor**

Memory

- **Flash (8 MB)**
  This is the program memory. The location is pointed to by the reset vector.
- **On-chip RAM (200 KB)**
  This is the location pointed to by the exception vector.
- **SDRAM (128 MB)**
  This is the data memory. Stack and heap memory are located here.

Timers

- **Timer (1ms)**
  This is used by the FreeRTOS Kernel and not available for use in the application software.
- **Timer (1µs)**

Peripherals implemented using Parallel I/O (PIO)

- **Green LEDs**
- **Red LEDs**
- **Slide switches**
- **Push button**s
  Note: Only keys 1, 2 and 3 are available for use in application software. Key 0 is bound to hardware reset. Key 1 is mapped to bit 0 of the PIO, key 2 to bit 1, and so on. Implementation of falling edge triggered interrupt is enabled for the buttons.

  Refer to pio.c for a code example on how to use these components.

Other peripherals

- **JTAG UART**
  This is the standard IO of the system. For instance, printf instruction uses this IO stream.

- **Seven segment display**
  Refer to seven_seg.c for a code example.

- **Character LCD (*CHARACTER_LCD*)**

Table of Escape Sequences

Sequence	Meaning
BS ‘\b’	Move the cursor to the left by one character
CR ‘\r’	Move the cursor to the start of the current line
LF ‘\n’	Move the cursor to the start of the line and move it down one line
ESC ‘\x1B’	Start a VT100 control sequence
ESC [<y> ; <x> H	Move the cursor to the y, x position specified – positions are counted from the top left which is 1;1
ESC [K	Clear from current cursor position to end of line
ESC [2 J	Clear whole screen

Refer to character_lcd.c for a code example.

- **PS/2 controller**

  Keyboards can be used without initialization. Refer to http://www.computer-engineering.org/ps2keyboard/scancodes2.html for keyboard scan codes.

  Refer to ps2_kb.c for a code example on using the keyboard.

- **UART**

- Parity	:	None
- Data bits	:	8
- Stop bits	:	1
- Baud rate	:	115200

Refer to uart.c and uart_isr.c for code examples. PC programs such as Putty can be used to communicate with the DE2-115 board through UART.

VGA display

- **SRAM**

  Stores the background data for each displayed pixel.
- **Video pixel buffer DMA**

  Reads background information from SRAM. The component operates at the resolution of 640×480 pixels.
- **Video character buffer DMA**

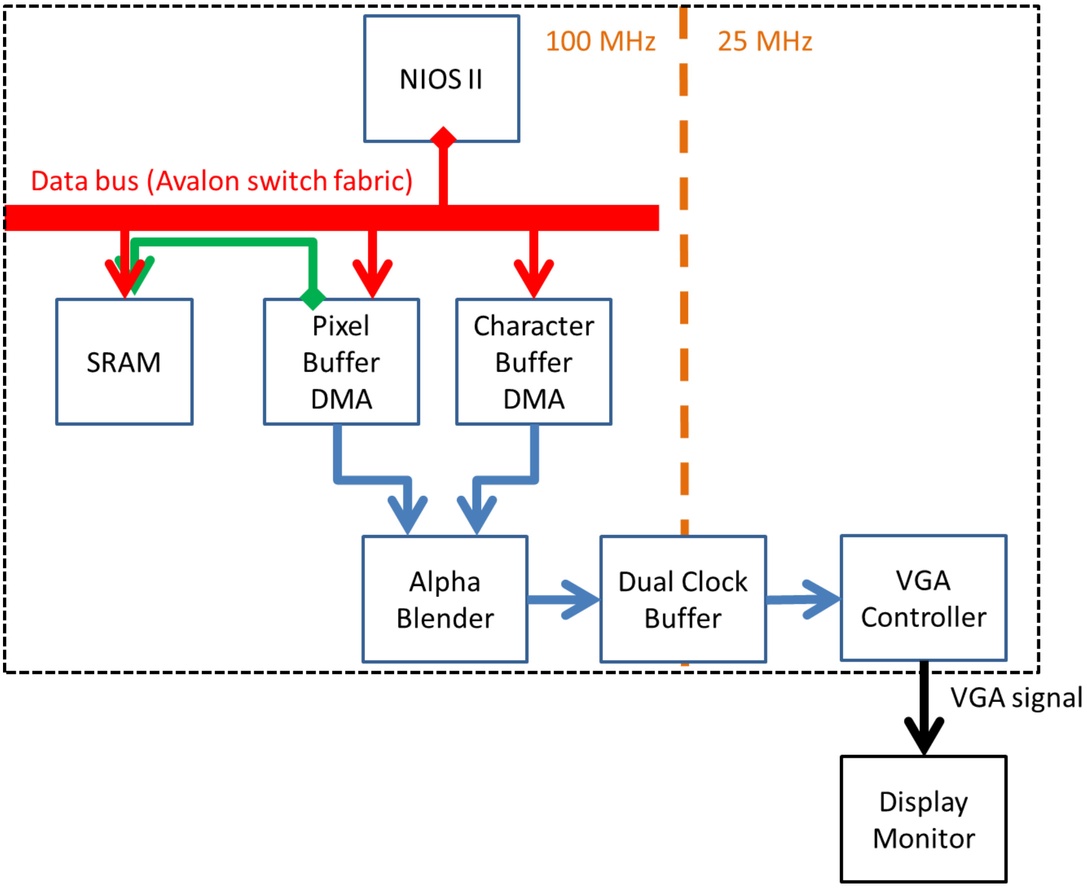
  Displays text in the foreground. Characters are positioned in an 80×60 grid.
- **Video alpha blender**

  Combines background and foreground.
- **Video VGA controller**

  Generate VGA display signals.

Character buffer DMA and pixel buffer DMA provides software libraries for drawing images.

Refer to char_buffer.c and pixel_buffer.c for code examples.



Frequency analyser (*FREQUENCY_ANALYSER*)

Accepts a series of digitised data of a signal (usually from an ADC) and determines the time period between two reference points of the signal. In this particular implementation the reference point is the maximum points of a sine wave.

This component generates an interrupt request when a reference point has been detected. The processor is able to read from the frequency analyser a value representing the number of ADC samples occurred between the last two reference points. The ADC sampling rate is set to 16 kHz. For example, if the number of ADC samples between two successive reference points is 324, the frequency of the sinusoidal wave is calculated as 16000 ÷ 324 = 49 Hz.

Refer to freq_relay.c for code examples.