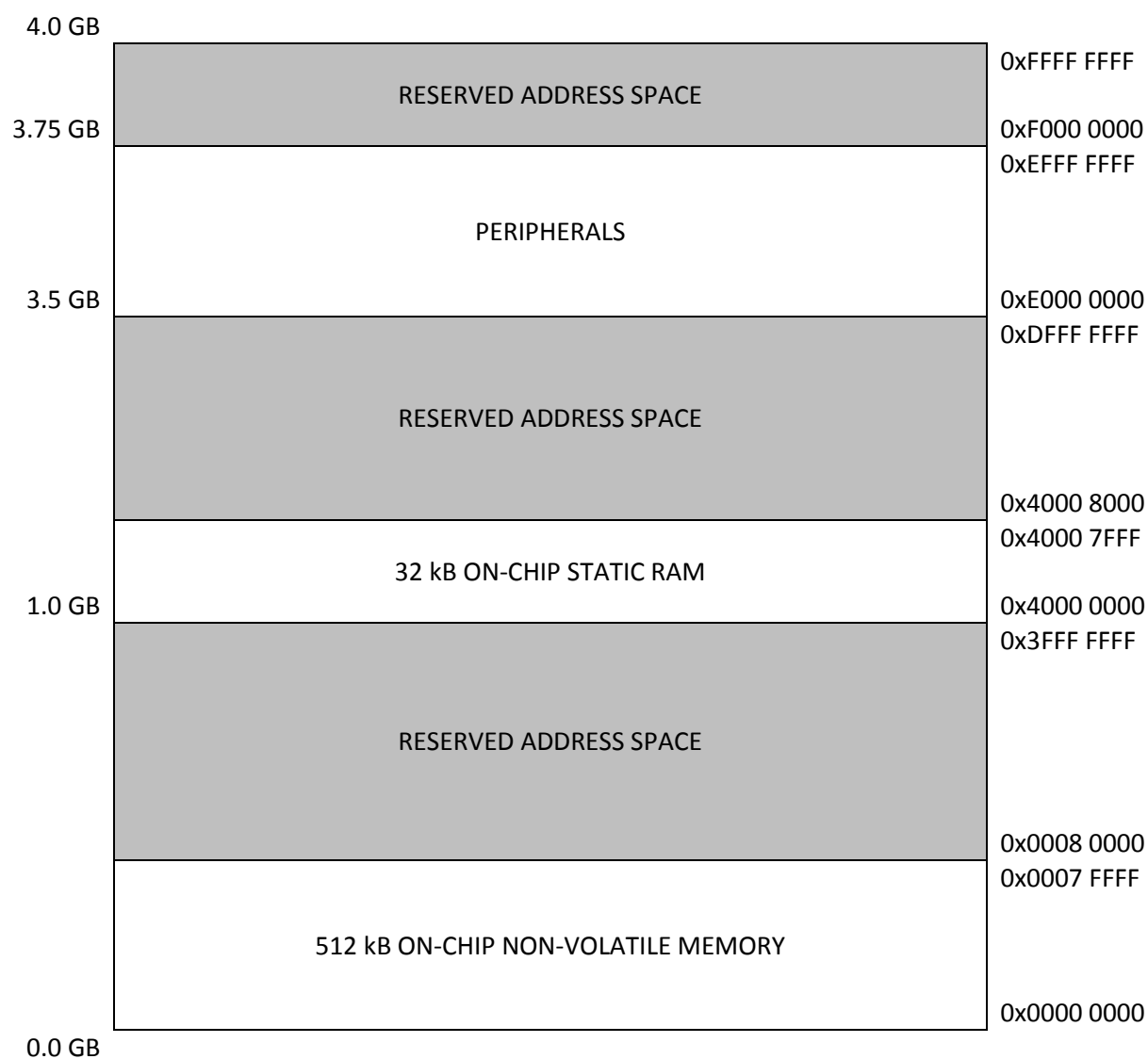


## 1.1 Memory Map of Development Board



## 1.2 Detail View of Peripheral Memory Map

3.75 GB

3.75 GB	NOT USED	0xEFFF FFFF
		0xE020 0000
	System Control Block	0xE01F C000
		0xE01F 8000
	NOT USED	
		0xE001 4000
	Interrupt Controller	0xE001 0000
	LCD	0xE000 C000
3.5 GB		
	LED0-8	0xE000 8000
	UART1	0xE000 4000
	UART0	0xE000 0000

## 1.3 Memory-Mapped Hardware Registers

### 1.3.1 UART0

The UARTs are 64-byte versions of the *standard* National Semiconductor UARTs. The model is 16750. For an in-depth explanation of the various registers and how to configure them, check out [this article](#).

<b>U0RBR</b>	<i>UART0 Receiver Buffer</i>	0xE000 0000
<b>U0THR</b>	<i>UART0 Transmitter Holding Buffer</i>	0xE000 0000
<b>U0DLL</b>	<i>UART0 Devisor Latch Low Byte</i>	0xE000 0000
<b>U0DLH</b>	<i>UART0 Devisor Latch High Byte</i>	0xE000 0004
<b>U0IER</b>	<i>UART0 Interrupt Enable Register</i>	0xE000 0004
<b>U0IIR</b>	<i>UART0 Interrupt Identification Register</i>	0xE000 0008
<b>U0FCR</b>	<i>UART0 FIFO Control Register</i>	0xE000 0008
<b>U0LCR</b>	<i>UART0 Line Control Register</i>	0xE000 000C
<b>U0LSR</b>	<i>UART0 Line Status Register</i>	0xE000 0014
<b>U0SCR</b>	<i>UART0 Scratch Register</i>	0xE000 001C

### 1.3.2 UART1

The UARTs are 64-byte versions of the *standard* National Semiconductor UARTs. The model is 16750. For an in-depth explanation of the various registers and how to configure them, check out [this article](#).

<b>U1RBR</b>	<i>UART1 Receiver Buffer</i>	0xE000 4000
<b>U1THR</b>	<i>UART1 Transmitter Holding Buffer</i>	0xE000 4000
<b>U1DLL</b>	<i>UART1 Devisor Latch Low Byte</i>	0xE000 4000
<b>U1DLH</b>	<i>UART1 Devisor Latch High Byte</i>	0xE000 4004
<b>U1IER</b>	<i>UART1 Interrupt Enable Register</i>	0xE000 4004
<b>U1IIR</b>	<i>UART1 Interrupt Identification Register</i>	0xE000 4008
<b>U1FCR</b>	<i>UART1 FIFO Control Register</i>	0xE000 4008
<b>U1LCR</b>	<i>UART1 Line Control Register</i>	0xE000 400C
<b>U1LSR</b>	<i>UART1 Line Status Register</i>	0xE000 4014
<b>U1SCR</b>	<i>UART1 Scratch Register</i>	0xE000 401C

### 1.3.3 LCD

The LCD is modeled after the Hitachi HD44780, although a bit simplified. Check out the *Instruction Description* in the HD44780.pdf file and then look at the provided assembler and C example code. It should be pretty self explanatory then. The exposed HW registers are:

<b>LCDC_IOCTL</b>	<i>LCD I/O Control Register</i>	0xE000 C000
<b>LCDC_DATA</b>	<i>LCD Data Transfer Register</i>	0xE000 C001

### 1.3.4 Programmable Interrupt Controller

The PIC is modeled after the PIC used in Samsung's S3C4510B microcontroller (also known as KS32C50100). For a detailed description of the exposed HW registers and how to configure the PIC, take a look at the S3C4510B Datasheet, Chapter 13 (Interrupt Controller).

<b>INTMOD</b>	<i>Interrupt Mode Register</i>	0xE001 0000
<b>INTPND</b>	<i>Interrupt Pending Register</i>	0xE001 0004
<b>INTMSK</b>	<i>Interrupt Mask Register</i>	0xE001 0008
<b>INTPRI0</b>	<i>Interrupt Priority Register 0</i>	0xE001 000C
<b>INTPRI1</b>	<i>Interrupt Priority Register 1</i>	0xE001 0010
<b>INTPRI2</b>	<i>Interrupt Priority Register 2</i>	0xE001 0014
<b>INTPRI3</b>	<i>Interrupt Priority Register 3</i>	0xE001 0018
<b>INTPRI4</b>	<i>Interrupt Priority Register 4</i>	0xE001 001C
<b>INTPRI5</b>	<i>Interrupt Priority Register 5</i>	0xE001 0020
<b>INTOFFSET</b>	<i>Interrupt Offset Register</i>	0xE001 0024
<b>INTOSET_FIQ</b>	<i>FIQ Interrupt Offset Register</i>	0xE001 0028
<b>INTOSET_IRQ</b>	<i>IRQ Interrupt Offset Register</i>	0xE001 002C
<b>INTPNDPRI</b>	<i>Interrupt Pending By Priority Register</i>	0xE001 0030
<b>INTPNDTST</b>	<i>Interrupt Pending Test Register</i>	0xE001 0034