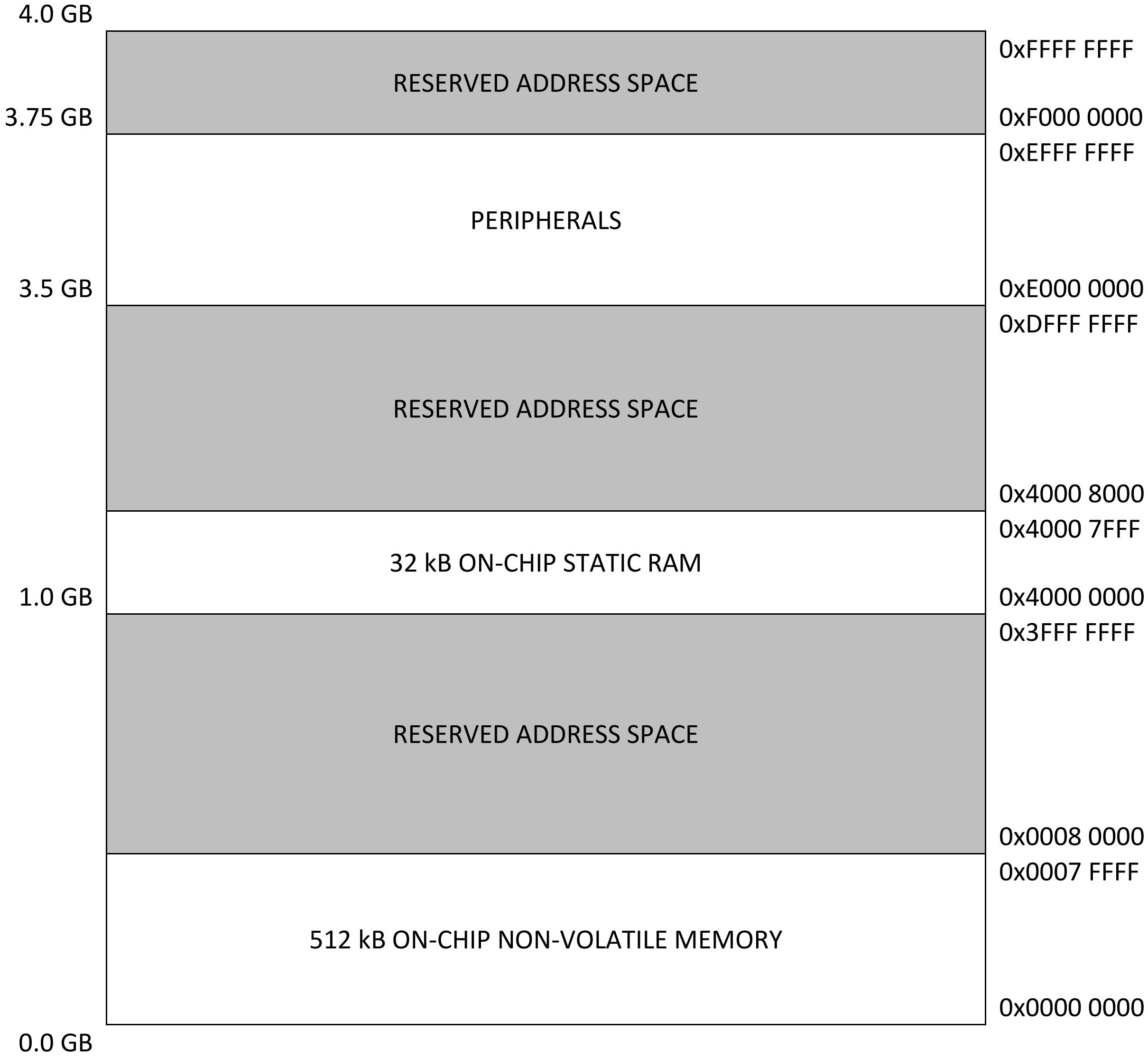
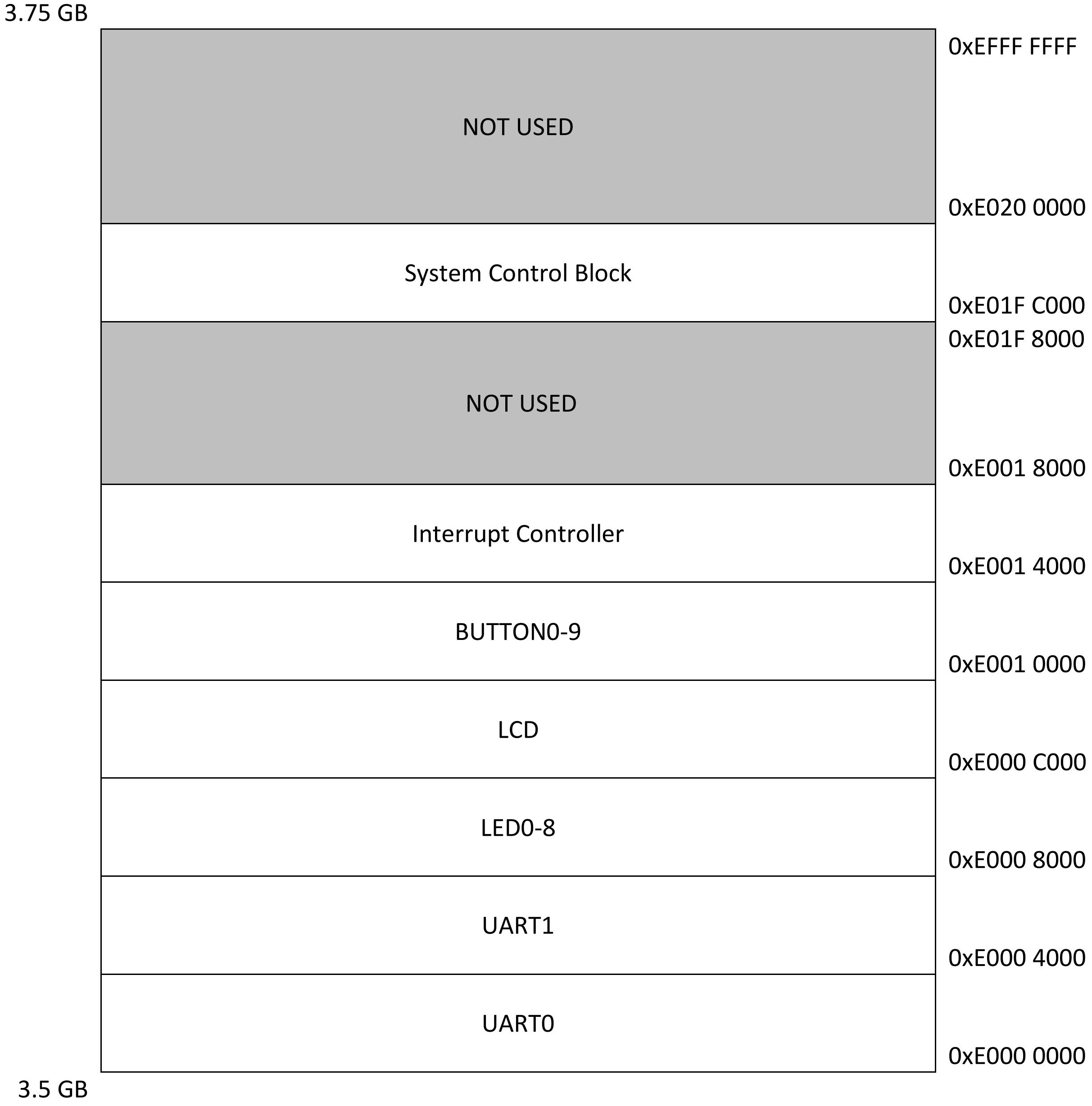
## 1.1 Memory Map of Development Board



## 1.2 Detail View of Peripheral Memory Map



## 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.

| | | |
|---|---|---|
| **U0RBR** | *UART0 Receiver Buffer* | `0xE000 0000` |
| **U0THR** | *UART0 Transmitter Holding Buffer* | `0xE000 0000` |
| **U0DLL** | *UART0 Devisor Latch Low Byte* | `0xE000 0000` |
| **U0DLH** | *UART0 Devisor Latch High Byte* | `0xE000 0004` |
| **U0IER** | *UART0 Interrupt Enable Register* | `0xE000 0004` |
| **U0IIR** | *UART0 Interrupt Identification Register* | `0xE000 0008` |
| **U0FCR** | *UART0 FIFO Control Register* | `0xE000 0008` |
| **U0LCR** | *UART0 Line Control Register* | `0xE000 000C` |
| **U0LSR** | *UART0 Line Status Register* | `0xE000 0014` |
| **U0SCR** | *UART0 Scratch Register* | `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.

| | | |
|---|---|---|
| **U1RBR** | *UART1 Receiver Buffer* | `0xE000 4000` |
| **U1THR** | *UART1 Transmitter Holding Buffer* | `0xE000 4000` |
| **U1DLL** | *UART1 Devisor Latch Low Byte* | `0xE000 4000` |
| **U1DLH** | *UART1 Devisor Latch High Byte* | `0xE000 4004` |
| **U1IER** | *UART1 Interrupt Enable Register* | `0xE000 4004` |
| **U1IIR** | *UART1 Interrupt Identification Register* | `0xE000 4008` |
| **U1FCR** | *UART1 FIFO Control Register* | `0xE000 4008` |
| **U1LCR** | *UART1 Line Control Register* | `0xE000 400C` |
| **U1LSR** | *UART1 Line Status Register* | `0xE000 4014` |
| **U1SCR** | *UART1 Scratch Register* | `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 explainatory then. The exposed HW registers are:

| | | |
|---|---|---|
| **LCDC_IOCTL** | *LCD I/O Control Register* | 0xE000 C000 |
| **LCDC_DATA** | *LCD Data Transfer Register* | 0xE000 C001 |

### 1.3.4 BUTTON0-9

The status of the Push Buttons can be queried by polling the respective bit of the Button Status Register. If the button is pressed the bit will be set, otherwise it will be cleared. The least significant bit (LSB) corresponds to button 0.

| | | |
|---|---|---|
| **BTNSTATUS** | *Button Status Register* | 0xE001 0000 |

### 1.3.5 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).

| | | |
|---|---|---|
| **INTMOD** | *Interrupt Mode Register* | 0xE001 4000 |
| **INTPND** | *Interrupt Pending Register* | 0xE001 4004 |
| **INTMSK** | *Interrupt Mask Register* | 0xE001 4008 |
| **INTPRI0** | *Interrupt Priority Register 0* | 0xE001 400C |
| **INTPRI1** | *Interrupt Priority Register 1* | 0xE001 4010 |
| **INTPRI2** | *Interrupt Priority Register 2* | 0xE001 4014 |
| **INTPRI3** | *Interrupt Priority Register 3* | 0xE001 4018 |
| **INTPRI4** | *Interrupt Priority Register 4* | 0xE001 401C |
| **INTPRI5** | *Interrupt Priority Register 5* | 0xE001 4020 |
| **INTOFFSET** | *Interrupt Offset Register* | 0xE001 4024 |
| **INTOSET_FIQ** | *FIQ Interrupt Offset Register* | 0xE001 4028 |
| **INTOSET_IRQ** | *IRQ Interrupt Offset Register* | 0xE001 402C |
| **INTPNDPRI** | *Interrupt Pending By Priority Register* | 0xE001 4030 |
| **INTPNDTST** | *Interrupt Pending Test Register* | 0xE001 4034 |

### 1.3.6 System Control Block

The System Control Block includes several system features and control registers for a number of functions that are not related to specific peripheral devices.

**PCON** *Power Control Register* `0xE01F C000`

Setting the least significant bit (LSB) of the *Power Control Register* will power down the development board and return control to the calling JavaScript code.