

## The On-Chip SRAM Configuration

\*\*\*\*\*

### 1. Introduction

\*\*\*\*\*

ADSP21535 has on-chip L1 and L2 SRAM. The L1 SRAM can be configured as data and instruction cache. The L1 Scratchpad SRAM and L2 SRAM can be used as a general on-chip memory which has low latency.

\*\*\*\*\*

### 2. system cache configuration

\*\*\*\*\*

Configuration for Eagle board:

| memory map     | start address | size | cacheable |
|----------------|---------------|------|-----------|
| MMR            | 0xFFC00000    | 4M   | No        |
| L1 Scratchpad  | 0xFFB00000    | 4K   | No        |
| L2 SRAM        | 0xF0000000    | 256K | No        |
| External SDRAM | 0x00000000    | 32M  | Yes       |
| PCI Memory     | 0xE7000000    | 16M  | No        |
| PCI IO/Config  | 0xEEF00000    | 1M   | No        |

Configuration for Hawk board:

| memory map     | start address | size | cacheable |
|----------------|---------------|------|-----------|
| MMR            | 0xFFC00000    | 4M   | No        |
| L1 Scratchpad  | 0xFFB00000    | 4K   | No        |
| L2 SRAM        | 0xF0000000    | 256K | No        |
| External SDRAM | 0x00000000    | 32M  | Yes       |
| PCI Memory     | 0xE0000000    | 16M  | No        |
| PCI IO/Config  | 0xEEF00000    | 1M   | No        |

#### NOTE:

- 1) Due to the DCPLB's limitation (there is only 16 DCPLB registers) only 16M space is reserved for PCI space.  
If data cache is NOT configured (CONFIG\_BLKFIN\_DCACHE not defined), all of the 128M PCI space can be accessed.
- 2) The Eagle board's configuration for PCI Memory is different with the one of Hawk board.  
The start address of PCI Memory in Eagle board is: 0xE7000000, the one of Hawk board is: 0xE0000000.

The reason is that the PCI BIOS (frio\_bios32.c) is always trying to allocate the PCI Memory for PCI device from high end first.

\*\*\*\*\*

### 3. On-Chip L1 Scratchpad SRAM Management

\*\*\*\*\*

To manage the L1 scratchpad SRAM, three interface functions are provided.

(1) void llsram\_init(void)

This function is used to initialize the L1 scratchpad memory

management. It should be called before using the L1 scratchpad SRAM.

(2) unsigned long l1sram\_alloc(unsigned long size)

This function is used to allocate the L1 scratchpad SRAM. The input argument is allocated memory size.

If memory allocation is successful, this function returns the allocated memory start address which is a non-zero value.

If the memory allocated failed, the function returns zero.

(3) int l1sram\_free(unsigned long saddr)

This function is used to free the allocated L1 scratchpad SRAM.

The argument saddr is the start address of the free memory, and the saddr should be the return value of the l1sram\_alloc()

function.

If the function is successful, the returned value is zero, otherwise, the returned value is -1.

\*\*\*\*\*

#### 4. On-Chip L1 Scratchpad SRAM Management

\*\*\*\*\*

To manage the L2 SRAM, three interface functions are provided.

(1) void l2sram\_init(void)

This function is used to initialize the L2 SRAM memory. It should be called before using the L2 SRAM.

(2) unsigned long l2sram\_alloc(unsigned long size)

This function is used to allocate the L2 SRAM. The input argument is the allocated memory size.

If the memory allocate successful, this function returns the allocated memory start address which is a non-zero value.

If memory allocation fails, the function returns a zero.

(3) int l2sram\_free(unsigned long saddr)

This function is used to free the allocated L2 SRAM memory.

The argument saddr is the start address of the free memory, and the saddr should be the return value of the l2sram\_alloc()

function.

If the function is successful the return value is zero, otherwise, the return value is -1.

\*\*\*\*\*

#### 5. Caution

\*\*\*\*\*