AppNote 6 - SpiNNaker Serial ROM Notes

SpiNNaker Group, School of Computer Science, University of Manchester

Steve Temple - 10 Dec 2011 - Version 1.00

Introduction

SpiNNaker chips contain an on-chip boot ROM which provides bootstrap facilities for the chip via Ethernet or a SpiNNaker link. In addition, the boot ROM provides an interface to an off-chip serial ROM (SROM). This may be used to provide an alternative start-up program or to initialise some SpiNNaker memory before the main part of the on-chip boot code is executed.

The SROM interface code occurs early in the on-chip ROM code. This is partly to maximise the opportunity to correct defective code in the on-chip ROM. After reset, a SpiNNaker chip will attempt to read from an external SROM. A specific data pattern is expected at the start of the SROM and if this is not found it is assumed that no SROM is present. If the pattern is found, the SpiNNaker chip loads data from the SROM according to a protocol that is defined in this document.

Hardware Interface

A standard SPI serial ROM is used and is controlled by the SpiNNaker general purpose I/O (GPIO) port. This port uses a 1.8V supply so the SROM must do likewise (or be level-shifted). The Microchip part 25AA1024 is suitable. An SPI device has a 4 wire interface. *NCS* is an active-low chip select. *SCK* is a clock and *SI* and *SO* are data input and output respectively. These are mapped on to GPIO bits as follows.

GPIO	Dir	SPI Function
5	Out	NCS - chip select
4	Out	SCK - clock
3	Out	SI - data to SROM
2	In	SO - data from SROM

SROM Data Format

The type of serial ROM used for SpiNNaker contains a sequence of bytes addressed upwards from 0. The bytes are organised into pages of 256 bytes for the purposes of erasing and writing. For use in SpiNNaker, data in the SROM is formed into variable length *blocks*. Zero or more *pad* bytes (0x55) may be placed before or after a block.

Each block begins with a *start* byte (0x3a). This is followed by a *length* halfword (2 bytes) denoting lengths from 0 to 65535. The length is followed by an *address* word (4 bytes) and this is followed by the number of *data* words (4 bytes) specified in the length. The block may then be followed by another block (with optional pad bytes) or there may be no more blocks. The absence of a block is signalled by a byte which is neither *start* (0x3a) nor *pad* (0x55).

In contrast to much of SpiNNaker, the length and data field are stored in Big Endian format. An example, showing a SROM containing two blocks is shown below.

The first block contains 7 words of data while the second one contains no data. The byte 0xFF marks the end of the blocks.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	55	3a	Length=7 00	07	Address = 0xf5007fe0 f5	00	7f	e0	Data 1				Data 2			
10	Data 3				Data 4				Data 5				Data 6			
20	Data 7				55	3a	Length=0 00	00	Address = 0x00007fe0 00	00	7f	e0	FF			
30																

Serial ROM Variants

There are a number of different serial ROMs which may be used with SpiNNaker. The main difference is in the capacity and this has an effect on the serial protocol which is used to read the SROM. Smaller SROMs of up to 64kB use a two byte address while larger SROMs use a three byte address. SpiNNaker assumes that a larger SROM is being used and always puts out a 3-byte read address when it starts to read the SROM. The smaller SROM which expects a 2-byte address can still be used provided that the data begins with a *pad* byte. It is recommended that a pad byte is *always* placed at the start of the SROM.

SROM Load Protocol

The code in the SpiNNaker boot ROM is executed by all CPUs in a SpiNNaker chip when it comes out of reset. It is not sensible for them all to attempt to read the SROM so a single, preselected CPU attempts to read the SROM and the remaining CPUs busy-wait until the handling of the SROM is complete.

The SROM is read block-by-block beginning at address zero. If a block contains a non-zero length, the data words in that block are loaded into Spinnaker memory starting at the word specified in the address field (which is a byte address). If a block contains a length of zero, the Spinnaker CPU which is performing the SROM load executes a branch and link (BL) to the address in the address field.

If the code which is branched to is APCS compliant (in particular, it must preserve r5-r7)) then if it returns (via the link register), code loading from the serial ROM will resume where it left off.

SROM data block

As a software convention, all SpiNNaker chips reserve an area of System RAM for 32 bytes of data copied from a serial ROM (if one is attached). The memory area is at 0xf5007fe0 (the top 32 bytes of System RAM). This area contains a number of fields as defined by this struct.

```
typedef struct srom_data       // Contents of SV SROM area (32 bytes)
{
  ushort flags;                // Various flags bits
  uchar mac_addr[6];           // MAC address of this chip
  uchar ip_addr[4];            // IP address of this chip
  uchar gw_addr[4];            // Gateway address
  uchar net_mask[4];           // Net mask
  ushort udp_port;             // UDP port for SCP commands
  ushort pad0;                 // Not used
```

```
    uint   pad1;               // Not used
    uint   pad2;               // Not used
  } srom_data_t;
```

The *flags* field must have its top bit set to indicate that data in the area has been loaded from serial ROM (it is set to zero before the on-chip ROM attempts to read the serial ROM). A hex dump of a serial ROM to set up the following parameters is shown below.

```
Flag: 8081
MAC:  00:00:a4:00:3e:0e
IP:   130.88.193.136
GW:   130.88.192.250
NM:   255.255.0.0
Port: 17893

00000000 55 3a 00 08 f5 00 7f e0 00 00 80 81 0e 3e 00 a4
00000010 88 c1 58 82 fa c0 58 82 00 00 ff ff 00 00 45 e5
00000020 00 00 00 00 00 00 00 00 aa
```

Change log:

- *1.00 - 25nov11- ST* - initial release - comments to *temples@cs.man.ac.uk*