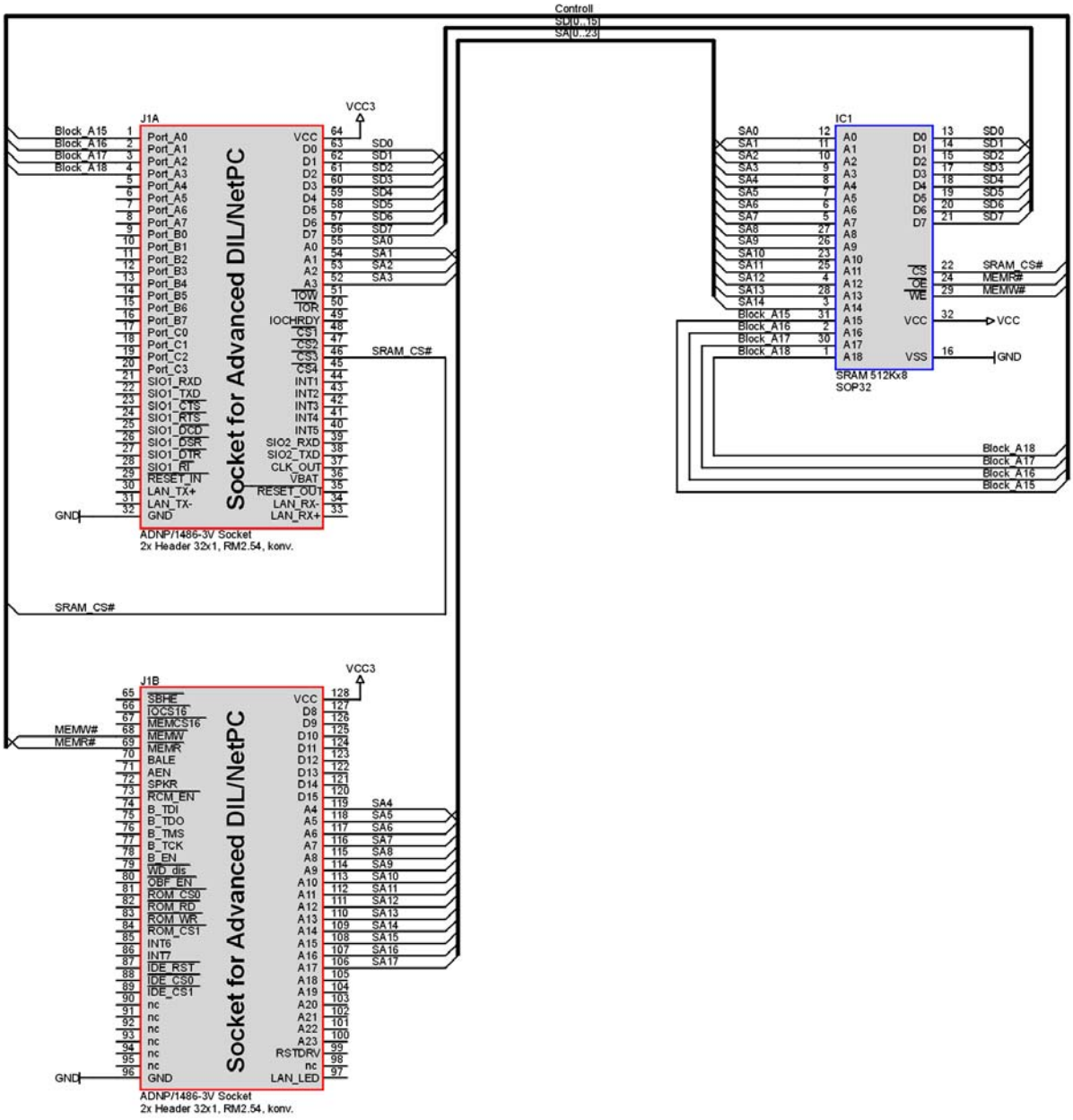
How to access external SRAM within the ADNP/1486 ISA address space

- ***1. Step***: Understand the following schematic. You can use any kind of SRAM chips with 8 bit data bus interfaces. The SRAM access goes over a 32 Kbyte memory window within the ADNP/1486 ISA address space. CS3 defines in this sample the base address of the memory windows. The SRAM address bits A0 to A14 are driven by the ADNP/1486 address bus. A15 and higher are connected to the ADNP/1486 PIO bits for bank switching.



- ***2. Step***: Download the following C sample code with the URL www.dilnetpc.com/mhtx8605.c and store the C file on your development system. Use this file for the first tests with your own hardware implementation of the SRAM interface.

- ***3. Step***: Understand the sample code. The sample code is using a 8-bit data bus to the external

SRAM. The SRAM chip select input is driven by CS3. The SRAM access window size is 32 Kbyte at base address 0xd000:0000.

```
// Demo user space SRAM access for DIL/NetPC ADNP/1486-3V
// Written by MHA - 14.12.2001
// You need SRAM at /CS3 with 8bit data bus size
// mmap function

#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/io.h>
#include <sys/mman.h>

#define I_REG 0x22
#define D_REG 0x23

/* interrupt control.. */

#define cli() __asm__ __volatile__ ("cli": : :"memory")
#define sti() __asm__ __volatile__ ("sti": : :"memory")

/*************************************************************\
 * Function....:  rdindex                                    *
 * Parameter...:  Adress of Index-Register to read           *
 * Return......:  Read Byte from respective register         *
 * Description.:  Reads index-register of DIL/NetPC          *
\*************************************************************/

static inline unsigned char rdindex(unsigned char cIndex)
{
   unsigned char cData, cOld_index;

   cli();                          // disable Ints
   cOld_index = inb(I_REG);
   outb(cIndex,I_REG);             // set indexregister
   cData=inb(D_REG);               // read data from indexregister
   outb(cOld_index,I_REG);
   sti();                          // enable Ints
   return(cData);
}

/*************************************************************\
 * Function....:  wrindex                                    *
 * Parameter...:  Adress of Index-Register to write to, value *
 * Description.:  Writes Byte to Index-Register of DIL/NetPC  *
\*************************************************************/
```

```
static inline void wrindex(unsigned char cIndex, unsigned char cData)
{

   unsigned char cOld_index;

   cli();                          // disable Ints
   cOld_index = inb(I_REG);
   outb(cIndex,I_REG);             // set indexregister
   outb(cData,D_REG);              // set data
   outb(cOld_index,I_REG);
   sti();                          // enable Ints
}

/*********************************************************************\
 * Function....:  DNP_chipsetting                                    *
 * Parameter...:  -                                                  *
 * Description.:  Enable SRAM for DIL/NetPC CS3 at 0xD0000 / 32 KB*
\*********************************************************************/

void DNP_chipsetting(void)
{
   // disable DIL/NetPC /CS3

   wrindex(0xa6,(rdindex(0xa6) & 0xf0) | 0x04);

   // DIL/NetPC /CS3 to output

   wrindex(0xa0,(rdindex(0xa0) & 0xcf) | 0x10);

   // disable DIL/NetPC /CS3

   wrindex(0xa6,(rdindex(0xa6) & 0xf0) | 0x04);

   // DIL/NetPC /CS3 = GPIO_CS2 = GP_CSC

   wrindex(0xb3,(rdindex(0xb3) & 0xf0) | 0x02);

   // DIL/NetPC /CS3 at 0xD0000 / 32 KB window

   wrindex(0xb9,(0xD0000>>18) & 0xff);

   // DIL/NetPC /CS3 at 0xD0000 / 32 KB window

   wrindex(0xba,((0xD0000>>14) & 0x0f) | 0xe0);

   // DIL/NetPC /CS3 8bit data bus size / Address decode only
```

```
   wrindex(0xbd,rdindex(0xbd) & 0xf0);

   // enable DIL/NetPC /CS3 (don`t disable CS1, CS2 and CS4 !!!)

   wrindex(0xa6,rdindex(0xa6) & 0xf0);
}

/*********************************************************************\
 * Function....:  main                                               *
 * Parameter...:  -                                                  *
 * Description.:  Read and write to DIL/NetPC SRAM                   *
\*********************************************************************/

int main(void)
{

   int iMemhdl, i;
   unsigned char *bMemptr;

   // clear screen

   printf("\33[2J\33[H");

   // get user identity

   if (geteuid() != 0) {
      printf("No root access rights !\n");
      exit(-1);
   }

   // full IO access permission

   iopl(3);

   printf("DIL/NetPC SRAM /CS3 mapping...");
   fflush(stdout);

   // do DIL/NetPC SRAM /CS3 mapping (SRAM at 0xD0000 / 32 KB)

   DNP_chipsetting();

   printf("done.\n\n");

   // open mem device for read/write

   iMemhdl = open("/dev/mem", O_RDWR);
   if (iMemhdl < 0) {
      printf("Can`t open /dev/mem !\n");
```

```
   exit(-1);
}

// get pointer to SRAM at 0xD0000-0xD7FFF

bMemptr = mmap((unsigned char *) 0,
               0x8000,
               PROT_READ | PROT_WRITE,
               MAP_SHARED,
               iMemhdl,
               0xD0000);

// error exit if no valid memory pointer

if (bMemptr <= 0){
   printf("mmap of /dev/mem fail !\n");
   exit(-1);
}

// display first 16 bytes from SRAM

printf("Display first 16 bytes of SRAM\n"
       "-----------------------------\n");

for (i=0; i<0x10; i++)
   printf("%02X ",*(bMemptr+i));
printf("\n\n");

// fill first 16 bytes from SRAM with 0x00

printf("Writing zero to first 16 bytes of SRAM\n");
for (i=0; i<0x10; i++)
   *(bMemptr+i) = 0;

// display first 16 bytes from SRAM

printf("Display first 16 bytes of SRAM after write\n"
       "-----------------------------------------\n");

for (i=0; i<0x10; i++)
   printf("%02X ",*(bMemptr+i));
printf("\n\n");

// fill first 16 bytes from SRAM with test pattern

printf("Fill first 16 bytes of SRAM with...\n");
for (i=0; i<0x10; i++)
   *(bMemptr+i) = i;
```

```
    // display first 16 bytes from SRAM

    printf("Display first 16 bytes of SRAM after write\n"
           "-----------------------------------------\n\n");

    for (i=0; i<0x10; i++)
    printf("%02X ",*(bMemptr+i));
    printf("\n\n");

    // unmap SRAM at 0xD0000-0xD7FFF

    munmap((unsigned char *) 0, 0x8000);

    // close mem device

    close(iMemhdl);
    return 0;
}
```

- ***4. Step***: Add the code for PIO handling. The sample code assumes that all PIO lines are 0. The PIO lines are used for bank switching.

Appendix 1: Pinout 128-pin QIL Connector ADNP/1486-3V (1. Part)

Pin	Name	Gruppe	Funktion
1	PA0	PIO	Parallel I/O, Port A, Bit 0
2	PA1	PIO	Parallel I/O, Port A, Bit 1
3	PA2	PIO	Parallel I/O, Port A, Bit 2
4	PA3	PIO	Parallel I/O, Port A, Bit 3
5	PA4	PIO	Parallel I/O, Port A, Bit 4
6	PA5	PIO	Parallel I/O, Port A, Bit 5
7	PA6	PIO	Parallel I/O, Port A, Bit 6
8	PA7	PIO	Parallel I/O, Port A, Bit 7
9	PB0	PIO	Parallel I/O, Port B, Bit 0
10	PB1	PIO	Parallel I/O, Port B, Bit 1
11	PB2	PIO	Parallel I/O, Port B, Bit 2
12	PB3	PIO	Parallel I/O, Port B, Bit 3
13	PB4	PIO	Parallel I/O, Port B, Bit 4
14	PB5	PIO	Parallel I/O, Port B, Bit 5
15	PB6	PIO	Parallel I/O, Port B, Bit 6
16	PB7	PIO	Parallel I/O, Port B, Bit 7
17	PC0	PIO	Parallel I/O, Port C, Bit 0
18	PC1	PIO	Parallel I/O, Port C, Bit 1
19	PC2	PIO	Parallel I/O, Port C, Bit 2
20	PC3	PIO	Parallel I/O, Port C, Bit 3
21	RXD	SIO	COM1 Serial Port, RXD Pin
22	TXD	SIO	COM1 Serial Port, TXD Pin
23	CTS	SIO	COM1 Serial Port, CTS Pin
24	RTS	SIO	COM1 Serial Port, RTS Pin
25	DCD	SIO	COM1 Serial Port, DCD Pin
26	DSR	SIO	COM1 Serial Port, DSR Pin
27	DTR	SIO	COM1 Serial Port, DTR Pin
28	RI	SIO	COM1 Serial Port, RI Pin
29	RESIN	RESET	RESET Input
30	TX+	LAN	10BASE-T Ethernet Interface, TX+ Pin
31	TX-	LAN	10BASE-T Ethernet Interface, TX- Pin
32	GND	----	Ground

Table 1.a: Pinout Pin 1 to 32

Appendix 1: Pinout 128-pin QIL Connector ADNP/1486-3V (2. Part)

Pin	Name	Gruppe	Funktion
33	RX+	LAN	10BASE-T Ethernet Interface, RX+ Pin
34	RX-	LAN	10BASE-T Ethernet Interface, RX- Pin
35	RESOUT	RESET	RESET Output
36	VBAT	PSP	SC410 Real Time Clock Battery Input
37	CLKOUT	PSP	Clock Output (Default 1.8432 MHz)
38	IRTXD	PSP	SC410 IrDA TXD Pin
39	IRRXD	PSP	SC410 IrDA RXD Pin
40	INT5	PSP	Programmable Interrupt Input 5
41	INT4	PSP	Programmable Interrupt Input 4
42	INT3	PSP	Programmable Interrupt Input 3
43	INT2	PSP	Programmable Interrupt Input 2
44	INT1	PSP	Programmable Interrupt Input 1
45	CS4	PSP	Programmable Chip Select Output 4
46	CS3	PSP	Programmable Chip Select Output 3
47	CS2	PSP	Programmable Chip Select Output 2
48	CS1	PSP	Programmable Chip Select Output 1
49	IOCHRDY	PSP	I/O Channel Ready
50	IOR	PSP	I/O Read Signal, I/O Expansion Bus
51	IOW	PSP	I/O Write Signal, I/O Expansion Bus
52	SA3	PSP	System Expansion Bus, Address Bit 3
53	SA2	PSP	System Expansion Bus, Address Bit 2
54	SA1	PSP	System Expansion Bus, Address Bit 1
55	SA0	PSP	System Expansion Bus, Address Bit 0
56	SD7	PSP	System Expansion Bus, Data Bit 7
57	SD6	PSP	System Expansion Bus, Data Bit 6
58	SD5	PSP	System Expansion Bus, Data Bit 5
59	SD4	PSP	System Expansion Bus, Data Bit 4
60	SD3	PSP	System Expansion Bus, Data Bit 3
61	SD2	PSP	System Expansion Bus, Data Bit 2
62	SD1	PSP	System Expansion Bus, Data Bit 1
63	SD0	PSP	System Expansion Bus, Data Bit 0
64	VCC	----	3.3 Volt Power Input

Table 1.b: Pinout Pin 33 to 64

Appendix 1: Pinout 128-pin QIL Connector ADNP/1486-3V (3. Part)

Pin	Name	Gruppe	Funktion
65	SBHE	PSP	System Byte High Enable, Sys.Exp.Bus
66	IOCS16	PSP	I/O Chip Select 16, Sys.Exp.Bus
67	MEMCS16	PSP	Memory Chip Select 16, Sys.Exp.Bus
68	MEMW	PSP	Memory Write Signal, Sys.Exp.Bus
69	MEMR	PSP	Memory Read Signal, Sys.Exp.Bus
70	BALE	PSP	Bus Latch Enable, Sys.Exp.Bus
71	AEN	PSP	Address Enable Signal, Sys.Exp.Bus
72	Reserved	PSP	Reserved. Don´t use.
73	RCME	PSP	Remote Console Mode Enable
74	Reserved	PSP	Reserved. Don´t use.
75	Reserved	PSP	Reserved. Don´t use.
76	Reserved	PSP	Reserved. Don´t use.
77	Reserved	PSP	Reserved. Don´t use.
78	Reserved	PSP	Reserved. Don´t use.
79	Reserved	PSP	Reserved. Don´t use.
80	Reserved	PSP	Reserved. Don´t use.
81	Reserved	PSP	Reserved. Don´t use.
82	Reserved	PSP	Reserved. Don´t use.
83	Reserved	PSP	Reserved. Don´t use.
84	Reserved	PSP	Reserved. Don´t use.
85	INT6	PSP	Programmable Interrupt Input 6
86	INT7	PSP	Programmable Interrupt Input 7
87	IDERES	PSP	IDE Interface Reset Output
88	IDECS0	PSP	IDE Interface Chip Select 0
89	IDECS1	PSP	IDE Interface Chip Select 1
90	Reserved	PSP	Reserved. Don´t use.
91	Reserved	PSP	Reserved. Don´t use.
92	Reserved	PSP	Reserved. Don´t use.
93	Reserved	PSP	Reserved. Don´t use.
94	Reserved	PSP	Reserved. Don´t use.
95	Reserved	PSP	Reserved. Don´t use.
96	GND	----	Ground

Table 1.c: Pinout Pin 65 to 96

Appendix 1: Pinout 128-pin QIL Connector DNP/1486-3V (4. Part)

Pin	Name	Gruppe	Funktion
97	LANLED	PSP	LAN Interface Activity LED
98	Reserved	PSP	Reserved. Don´t use.
99	RSTDRV	PSP	Reset Output, System Expansion Bus
100	SA23	PSP	System Expansion Bus, Address Bit 23
101	SA22	PSP	System Expansion Bus, Address Bit 22
102	SA21	PSP	System Expansion Bus, Address Bit 21
103	SA20	PSP	System Expansion Bus, Address Bit 20
104	SA19	PSP	System Expansion Bus, Address Bit 19
105	SA18	PSP	System Expansion Bus, Address Bit 18
106	SA17	PSP	System Expansion Bus, Address Bit 17
107	SA16	PSP	System Expansion Bus, Address Bit 16
108	SA15	PSP	System Expansion Bus, Address Bit 15
109	SA14	PSP	System Expansion Bus, Address Bit 14
110	SA13	PSP	System Expansion Bus, Address Bit 13
111	SA12	PSP	System Expansion Bus, Address Bit 12
112	SA11	PSP	System Expansion Bus, Address Bit 11
113	SA10	PSP	System Expansion Bus, Address Bit 10
114	SA9	PSP	System Expansion Bus, Address Bit 9
115	SA8	PSP	System Expansion Bus, Address Bit 8
116	SA7	PSP	System Expansion Bus, Address Bit 7
117	SA6	PSP	System Expansion Bus, Address Bit 6
118	SA5	PSP	System Expansion Bus, Address Bit 5
119	SA4	PSP	System Expansion Bus, Address Bit 4
120	SD15	PSP	System Expansion Bus, Data Bit 15
121	SD14	PSP	System Expansion Bus, Data Bit 14
122	SD13	PSP	System Expansion Bus, Data Bit 13
123	SD12	PSP	System Expansion Bus, Data Bit 12
124	SD11	PSP	System Expansion Bus, Data Bit 11
125	SD10	PSP	System Expansion Bus, Data Bit 10
126	SD9	PSP	System Expansion Bus, Data Bit 9
127	SD8	PSP	System Expansion Bus, Data Bit 8
128	VCC	----	3.3 Volt Power Input

Table 1.d: Pinout Pin 97 to 128