

MỞ RỘNG BỘ NHỚ NGOÀI

-----ΦΔΦ-----

Mở đầu:

Việc mở rộng bộ nhớ SRAM ngoài là rất cần thiết khi phải xử lý dữ liệu với dung lượng lớn và phép tính phức tạp. Trong bài này ta xem qua cách sử dụng bộ nhớ mở rộng như thế nào?

Các đề mục trong bài:

Các hình minh họa:.....	2
Các bảng chú thích:.....	2
A. Các vùng bộ nhớ của LPC2214:	3
B. Điều khiển vùng nhớ ngoài<External Memory Controller - EMC>:	3
B.I. Chức năng:.....	3
B.II. Giải thích:.....	4
II.1. Các chân chức năng:.....	4
II.2. Các thanh ghi chức năng:.....	4
II.3. Kết nối bộ nhớ ngoài:.....	5
II.4. Giảm đồ định thì truy cập bộ nhớ ngoài qua bus chuẩn:.....	7
B.III. Lựa chọn bộ nhớ ngoài:.....	8
C. Phân cứng thực hiện:	8
D. Cách truy cập RAM ngoài:	9
Tiến hành cụ thể:	10
E. Kết luận:	16

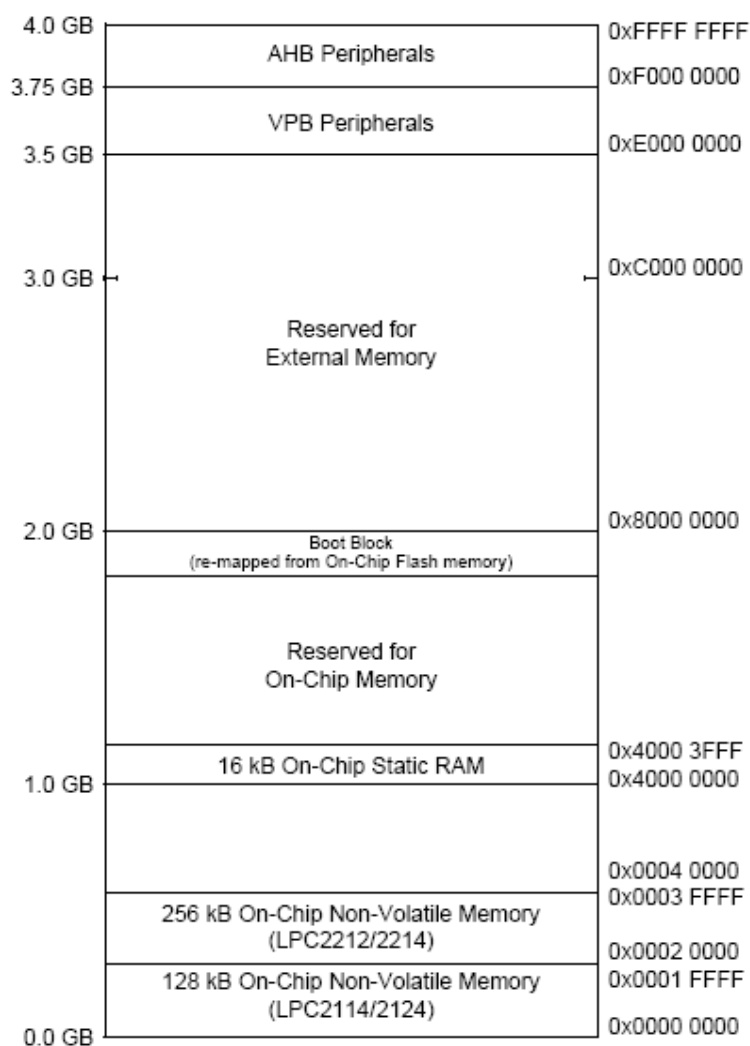
Các hình minh họa:

Hình 1: Các vùng bộ nhớ của LPC2214	3
Hình 2: Kết nối bank bộ nhớ ngoài 32 bit.....	6
Hình 3: Kết nối bộ nhớ ngoài 16 bit	6
Hình 4: Kết nối bộ nhớ ngoài 8 bit	7
Hình 5: Đọc từ bộ nhớ ngoài (Ví dụ với WST1=0 và WST1=1)	7
Hình 6: Ghi vào bộ nhớ ngoài (Ví dụ với WST1=0 và WST2=1).....	7
Hình 7: Sơ đồ kết nối bộ nhớ SRAM mở rộng (<i>Designed by Nguyen Thanh Phuc - Vitek</i>)	9
Hình 8: Lựa chọn cấu hình MCU cho giao tiếp SRAM	10
Hình 9: Lựa chọn cấu hình Half Word cho SRAM1 và SRAM2	11
Hình 10: Lựa chọn cấu hình Byte cho SRAM1	12
Hình 11: Kết quả mô phỏng bằng chương trình Keil-uV3	14
Hình 12: Kết quả thật sự với DATA=8 bit	14
Hình 13: Kết quả mô phỏng với DATA=32 bit-dữ liệu kiểu Little Endian.....	15
Hình 14: Kết quả thực tế với 32 bit data.....	15

Các bảng chú thích:

Bảng 1: Tầm địa chỉ của bộ nhớ ngoài	4
Bảng 2: Các chân chức năng điều khiển bộ nhớ ngoài.....	4
Bảng 3: Các thanh ghi chức năng điều khiển truy cập bộ nhớ ngoài	4
Bảng 4: Cấu hình bank thanh ghi.....	5
Bảng 5: Mặc định chiều rộng bộ nhớ sau khi Reset	5
Bảng 6: Bộ nhớ ngoài và các yêu cầu hệ thống.....	8

A. Các vùng bộ nhớ của LPC2214:



Hình 1: Các vùng bộ nhớ của LPC2214

Theo hình vẽ trên, ta thấy rằng địa chỉ vùng nhớ từ 0x8000 0000 đến 0xDFFF FFFF <1.5G> dành cho kết nối ngoại vi với bộ nhớ ngoài. Thực sự chỉ có 24 đường địa chỉ tích cực, nghĩa là được: $2^4 \cdot 2^{20} = 16M$. Định địa chỉ theo word, ta được: $16MWords = 64Mbytes$.

B. Điều khiển vùng nhớ ngoài <External Memory Controller - EMC>:

B.1. Chức năng:

- Hỗ trợ các thiết bị phân vùng bộ nhớ tĩnh bao gồm RAM, ROM, FLASH, burst ROM và một số thiết bị xuất/nhập khác.
- Hoạt động ở chế độ đọc định trang bất đồng bộ ở các hệ thống nhớ hợp lệ.
- Chế độ truy cập đọc burst ROM bất đồng bộ.
- Cấu hình độc lập cho 4 bank, mỗi bank 16Mbytes.
- Chu kì bus rỗi có thể lập trình được <1-16MC>
- Trạng thái chờ đọc và ghi có thể lập trình được <tới 32 trạng thái> cho các thiết bị RAM tĩnh.
- Trạng thái chờ khởi tạo và tuần tự đọc burst ROM có thể lập trình được.
- Bảo vệ chống ghi có thể lập trình.
- Hoạt động ở chế độ burst có thể lập trình.
- Các bus dữ liệu có thể chọn ở 8/16/32 bits.
- Cho phép điều khiển đọc byte riêng lẻ.

B.II. Giải thích:

Điều khiển bộ nhớ tĩnh ngoài là modul phụ AMBA AHB đưa ra kết nối giữa hệ thống bus AMBA AHB và các thiết bị nhớ ngoài. Nó hỗ trợ cho 4 bank bộ nhớ có cấu hình độc lập một cách đồng thời. Mỗi bank bộ nhớ có thể là ROM, SRAM, Flash, Burst ROM hay các thiết bị xuất nhập khác.

Mỗi bank có thể cấu hình 8/16/32 bit.

Các đường địa chỉ A[23:0], giải mã địa chỉ A[25:24]. Vùng địa chỉ của bộ nhớ ngoài được phân vùng ở hình 2 <và bảng dưới đây>, trong đó bank 0 có thể được dùng để khởi tạo chương trình boot nếu các chân BOOT1:0=11

Bank	Tầm địa chỉ	Cấu hình cho thanh ghi
0	8000 0000 - 80FF FFFF	BCFG0
1	8100 0000 - 81FF FFFF	BCFG1
2	8200 0000 - 82FF FFFF	BCFG2
3	8300 0000 - 83FF FFFF	BCFG3

Bảng 1: Tầm địa chỉ của bộ nhớ ngoài

II.1. Các chân chức năng:

Kí hiệu	Loại	Chức năng
D[31:0]	Xuất/nhập	Đường dữ liệu của bộ nhớ ngoài
A[23:0]	Xuất	Đường địa chỉ của bộ nhớ ngoài.
OE	Xuất	Tín hiệu cho phép đọc (tích cực ở mức thấp)
BLS[3:0]	Xuất	Tín hiệu chọn từng Byte tích cực ở mức thấp.
WE	Xuất	Tín hiệu cho phép ghi tích cực ở mức thấp.
CS[3:0]	Xuất	Tín hiệu chọn chip(tích cực ở mức thấp)

Bảng 2: Các chân chức năng điều khiển bộ nhớ ngoài

II.2. Các thanh ghi chức năng:

Bộ điều khiển vùng nhớ ngoài có 4 thanh ghi chức năng

Tên	Chức năng	Truy cập	Giá trị sau khi Reset	Địa chỉ
BCFG0	Thanh ghi cấu hình cho bank 0 của bộ nhớ	Đọc/ghi	0x0000 FBEF	0xFFE00000
BCFG1	Thanh ghi cấu hình cho bank 1 của bộ nhớ	Đọc/ghi	0x2000 FBEF	0xFFE00004
BCFG2	Thanh ghi cấu hình cho bank 2 của bộ nhớ	Đọc/ghi	0x1000 FBEF	0xFFE00008
BCFG3	Thanh ghi cấu hình cho bank 3 của bộ nhớ	Đọc/ghi	0x0000 FBEF	0xFFE0000C

Bảng 3: Các thanh ghi chức năng điều khiển truy cập bộ nhớ ngoài

Mỗi thanh ghi chức năng như vậy điều khiển các thông tin:

- Số xung nhịp rỗi chèn giữa quá trình đọc và ghi trong cùng bank, hoặc giữa các quá trình truy cập giữa bank này và bank khác, tránh sự tranh chấp giữa các thiết bị.<1 đến 17 chu kì xung nhịp>
- Chiều dài của chu kì đọc, bỏ qua quá trình đọc theo sau từ burst ROM<3 tới 35 chu kì xung nhịp>
- Chiều dài của quá trình ghi <3 đến 19 chu kì xung nhịp>
- Bank thanh ghi này có chống ghi hay không?
- Sử dụng 8/16/32 bit dữ liệu

BCFG0-3	Tên	Chức năng	Giá trị Reset
3:0	IDCY	Điều khiển số chu kì xung nhịp ít nhất để EMC giữ giữa quá trình truy cập đọc và ghi, truy cập vào bank thanh ghi khác và bank thanh ghi này, nhằm tránh mọi tranh chấp có thể xảy ra. Số chu kì CCLK bỏ qua chính là giá trị của byte này +1.	1111
4	Dự trữ	Dự trữ, người lập trình không được ghi '1' vào bit này. Giá trị của bit này không có ý nghĩa.	NA
9:5	WST1	Chiều dài của chu kì đọc, bỏ qua quá trình đọc theo sau từ burst ROM<3 tới 35 chu kì xung nhịp> Số chu kì xung nhịp bỏ qua bằng giá trị của WST1+3	11111
10	RBLE	RBLE = 0 →byte hoặc <byte, BLS3:0 = 1111 suốt quá trình đọc. RBLE = 1 →16 bit hoặc word, BLS3:0 = 0000 suốt quá trình đọc.	0
15:11	WST2	-SRAM banks, trường này điều khiển chiều dài của quá trình ghi, bao gồm: • 1 chu kì CCLK định địa chỉ với CS, BLS, WE ở mức cao. • (WST2+1) CCLK địa chỉ hợp lệ với CS, BLS, và WE ở mức thấp. • 1 CCLK với địa chỉ hợp lệ, CS thấp, BLS và WE ở mức cao. -Burst ROM banks, trường này điều khiển chiều dài của quá trình truy cập sau đó, có chiều dài bằng chính nó+1 chu kì CCLK	11111
16:23	Dự trữ	Dành dự trữ, người lập trình không được ghi '1' vào các bit dự trữ. Các giá trị đọc được từ bit dự trữ không được định nghĩa	NA
24	BUSERR	Trường hợp EMC phát hiện rằng AMBA yêu cầu dữ liệu >32bit. ARM7TDMI-S không hỗ trợ những yêu cầu này.	0
25	WPERR	Bit này được bật khi phần mềm cố gắng ghi vào bank thanh ghi đã có bit bảo vệ chống ghi(WP=1)	0
26	WP	Bit bảo vệ chống ghi	0
27	BM	Bit nhận dạng burst-ROM bank.	0
29:28	MW	Điều khiển chiều dài của bus dữ liệu trên bank: o 00=8 bit, o 01=16 bit, o 10=32 bit, o 11=không được sử dụng	Xem bảng dưới
31:30	AT	AT=00	00

Bảng 4: Cấu hình bank thanh ghi

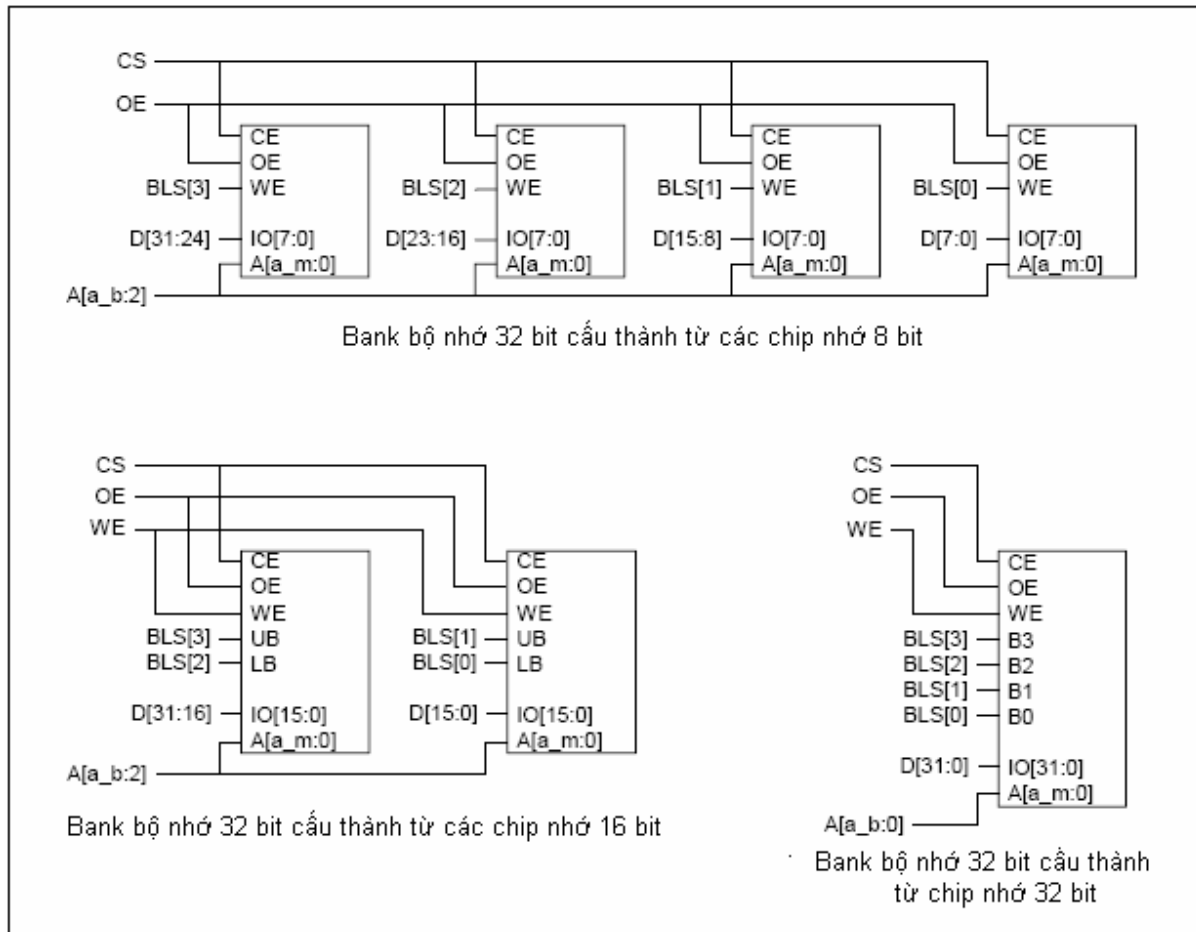
Bảng dưới đây cho biết các trạng thái của BCF G0[29:28] sau khi Boot loader hoạt động. Giá trị Reset cứng (Reset ngoài) mặc định của nó là 10.

Bank	BOOT[1:0] trong quá trình Reset	Giá trị Reset của BCFG[29:28]	Bề rộng của bộ nhớ
0	LL	00	8 bits
0	LH	01	16 bits
0	HL	10	32 bits
1	XX	10	32 bits
2	XX	01	16 bits
3	XX	00	8 bits

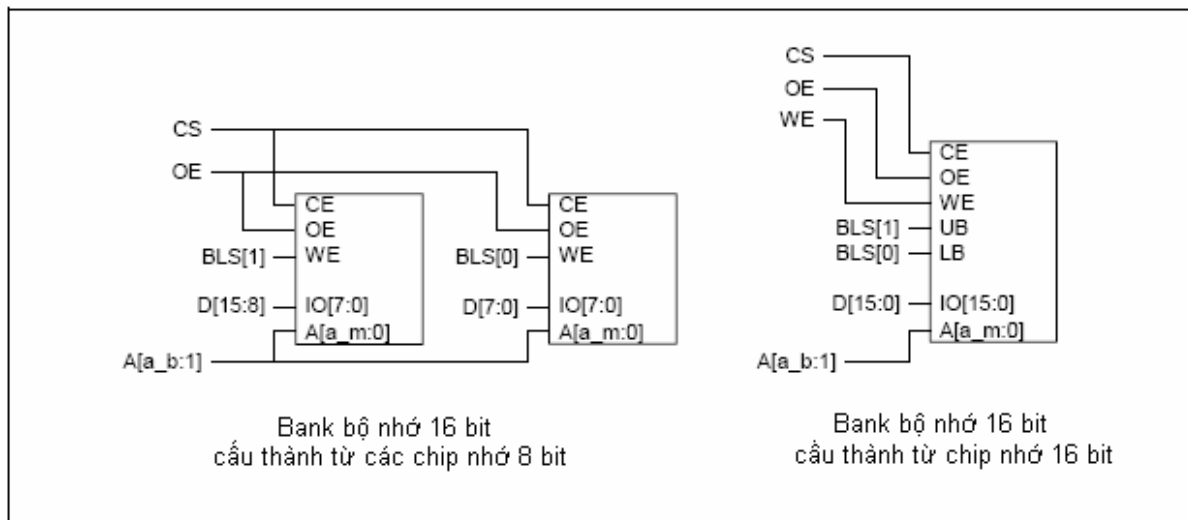
Bảng 5: Mặc định chiều rộng bộ nhớ sau khi Reset

II.3. Kết nối bộ nhớ ngoài:

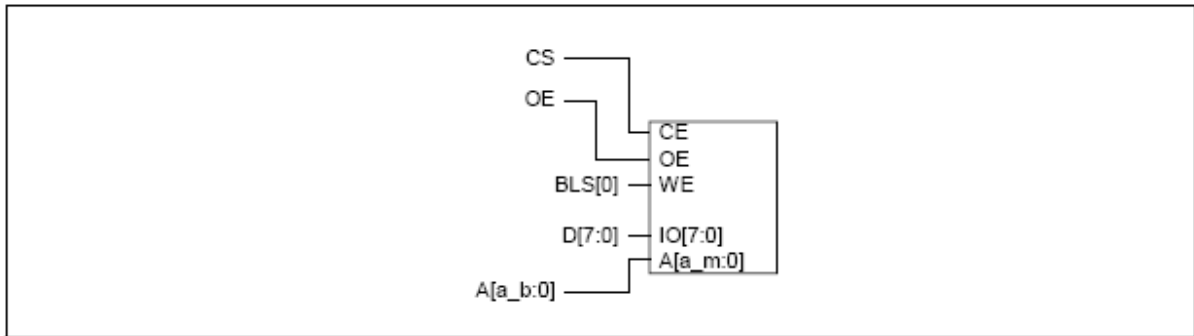
Kết nối bộ nhớ phụ thuộc bề rộng bank thanh ghi (32/16/8 được chọn thông qua bit MW ở thanh ghi BCFG). Hơn nữa, sự lựa chọn chip bộ nhớ đòi hỏi khai báo bit RBLE trên thanh ghi BCFG, RBLE=0→bộ nhớ ngoài dựa trên các chip nhớ 8 bit, RBLE = 1→bộ nhớ ngoài dựa trên các chip 16/32 bit. Nếu bank bộ nhớ được cấu hình là 32bit, đường địa chỉ A0 và A1 có thể được sử dụng như là các đường không địa chỉ. Bank bộ nhớ 16 bit không sử dụng A0, bank 8 bit bắt buộc đường địa chỉ tới A0. Cấu hình A1 và/hoặc A0 làm chức năng đường địa chỉ/không phải đường địa chỉ bằng các bit 23 và 24 trên thanh ghi lựa chọn chức năng (thanh ghi PINSEL2). Kí hiệu a_b trong hình sau đề cập đến các đường địa chỉ cao nhất trong bus dữ liệu. Kí hiệu a_m đề cập đến các đường địa chỉ cao nhất của bộ nhớ ngoài.



Hình 2: Kết nối bank bộ nhớ ngoài 32 bit



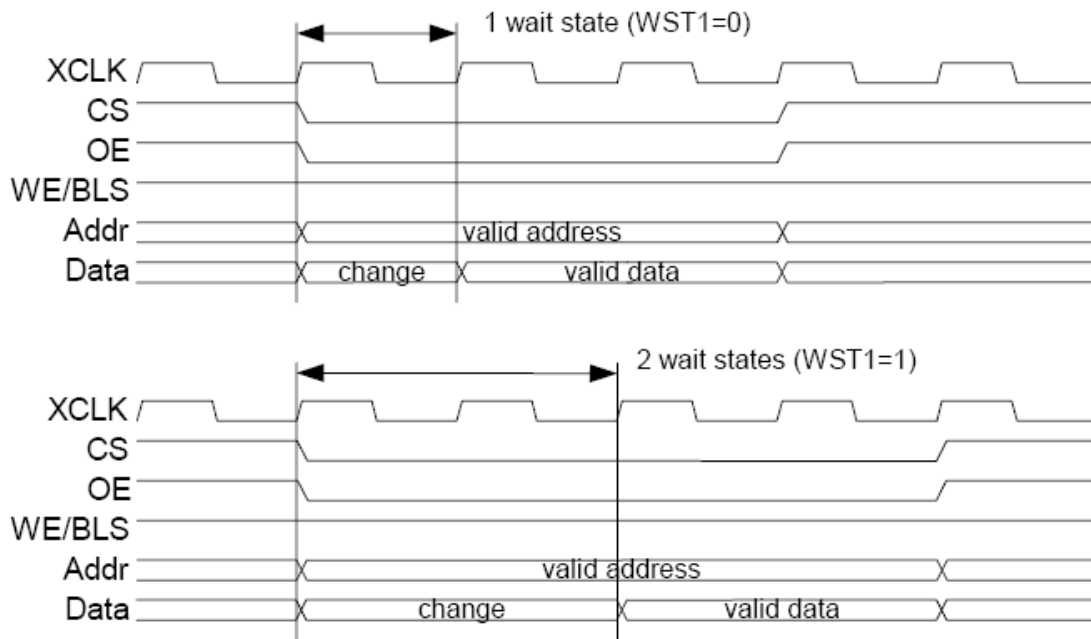
Hình 3: Kết nối bộ nhớ ngoài 16 bit



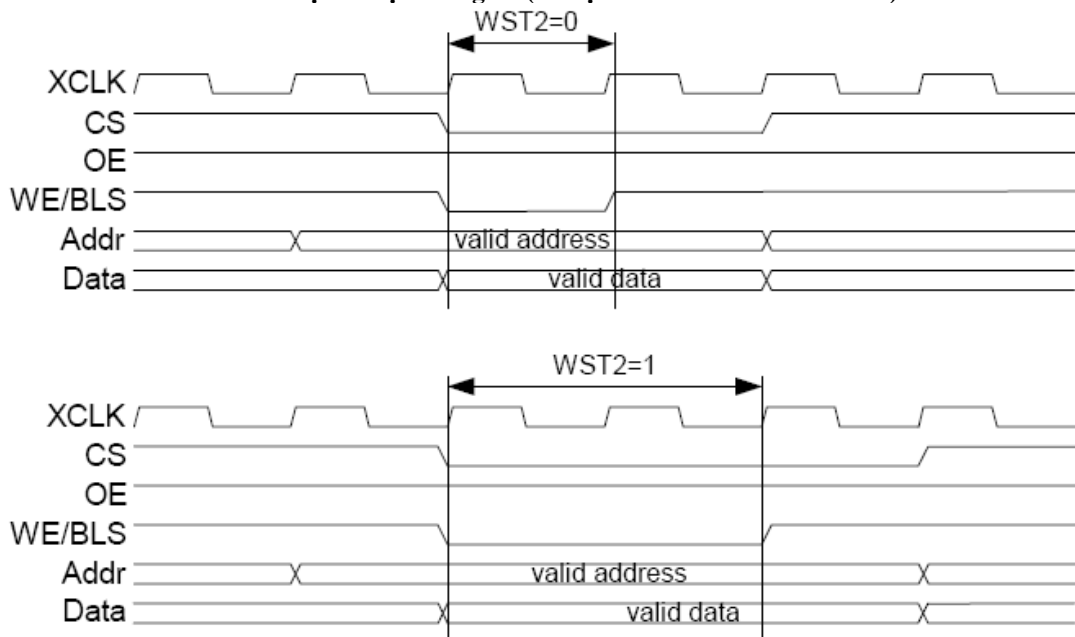
Hình 4: Kết nối bộ nhớ ngoài 8 bit

II.4. Giản đồ định thì truy cập bộ nhớ ngoài qua bus chuẩn:

Các giản đồ dưới đây cho ta thấy chu kỳ truy cập <xuất/nhập> dữ liệu ngoài. XCLK là tín hiệu xung nhịp trên chân P3.23. Trong các minh họa dưới đây, ta chọn chu kỳ XCLK=CCLK.



Hình 5: Đọc từ bộ nhớ ngoài (Ví dụ với WST1=0 và WST1=1)



Hình 6: Ghi vào bộ nhớ ngoài (Ví dụ với WST1=0 và WST2=1)

Hai hình trên minh họa cho cách đọc và ghi dữ liệu từ/vào bộ nhớ ngoài, trường hợp minh họa với WST1, WST2 xác định trước, tất nhiên, các trường hợp khác của WST1, WST2 cũng tương tự với lưu ý từ bảng 6.

B.III. Lựa chọn bộ nhớ ngoài:

Trong bảng chú thích dưới đây:

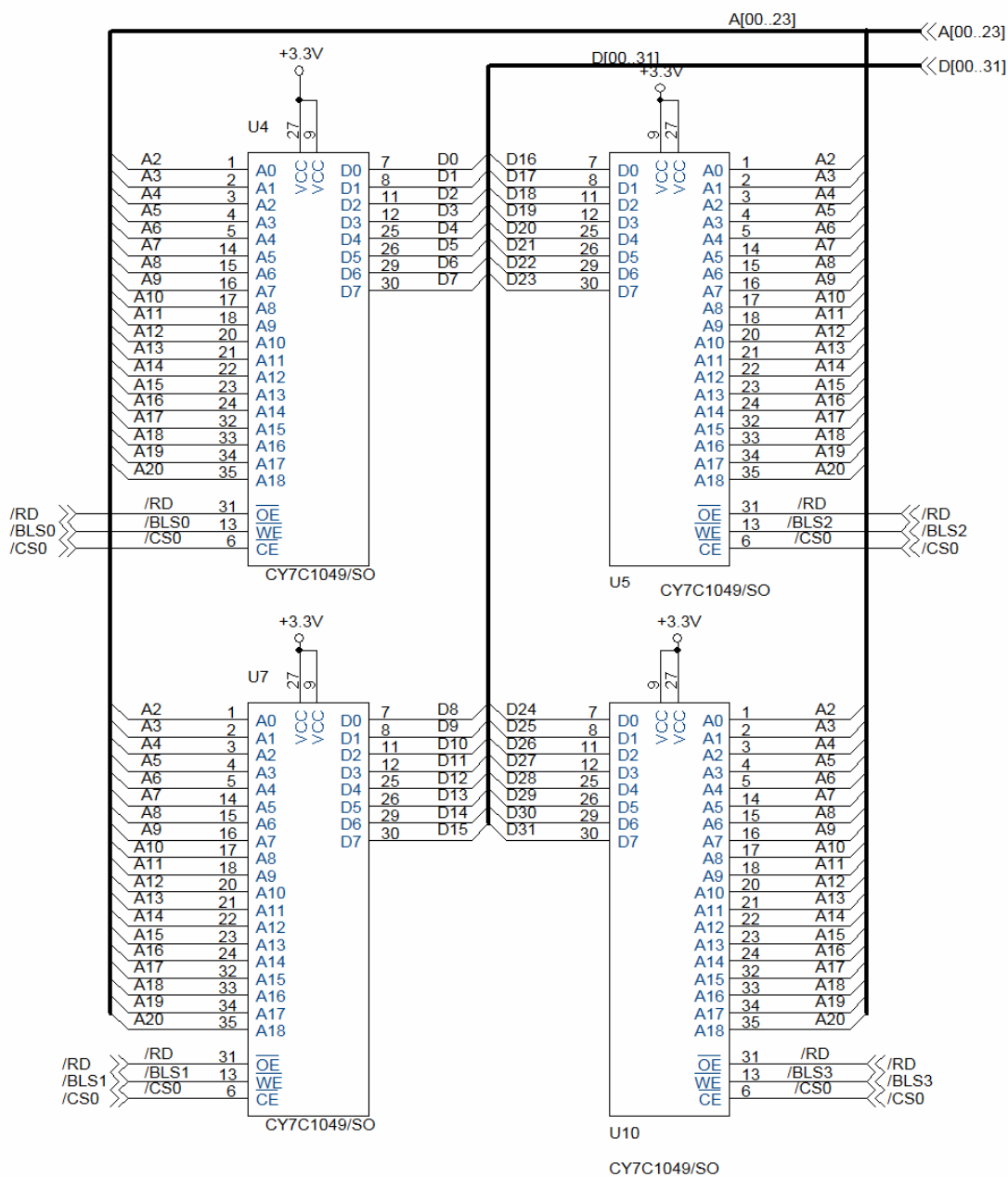
- ✚ t_{AA} =thời gian truy cập để đọc dữ liệu (read access times)
- ✚ t_{write} =thời gian truy cập để ghi dữ liệu (write access times)
- ✚ t_{cyc} =chu kỳ một chu kỳ XCLK.
- ✚ f_{max} =tần số CCLK lớn nhất của hệ thống có thể được khi hoạt động với bộ nhớ ngoài.

Chu kỳ truy cập	Tần số lớn nhất	WST (WST>=0; làm tròn lên thành số integer)	Thời gian truy cập bộ nhớ cần thiết
Đọc chuẩn	$f_{max} \leq \frac{2 + WST1}{t_{RAM} + 20ns}$	$f_{max} \geq \frac{t_{RAM} + 20ns}{t_{cyc}} - 2$	$t_{RAM} \leq t_{cyc} \times (2 + WST1) - 20ns$
Ghi chuẩn	$f_{max} \leq \frac{1 + WST2}{t_{RAM} + 5ns}$	$f_{max} \geq \frac{t_{WRITE} - t_{cyc} + 5ns}{t_{cyc}} - 2$	$t_{RAM} \leq t_{cyc} \times (1 + WST2) - 5ns$

Bảng 6: Bộ nhớ ngoài và các yêu cầu hệ thống

C. Phần cứng thực hiện:

Kết nối phần cứng sử dụng bank bộ nhớ 32 bit cấu thành từ các chip nhớ 8 bit. Hình mô phỏng:



Hình 7: Sơ đồ kết nối bộ nhớ SRAM mở rộng (Designed by Nguyen Thanh Phuc - Vitek)

Các đường địa chỉ A0-A1 không sử dụng trong cấu hình này.

D. Cách truy cập RAM ngoài:

Các đường địa chỉ từ A2-A20 định địa chỉ của vùng nhớ SRAM, ta có thể cho truy cập theo từng byte, half word hoặc word tùy cấu hình bit \overline{BLS}_i ($i = \{1, 2, 3\}$). Địa chỉ vùng nhớ ta truy cập có thể cấu hình thuộc 1 trong 4 bank thành ghi BCFG. Giả sử ta dùng bank 1: Tầm địa chỉ truy cập:

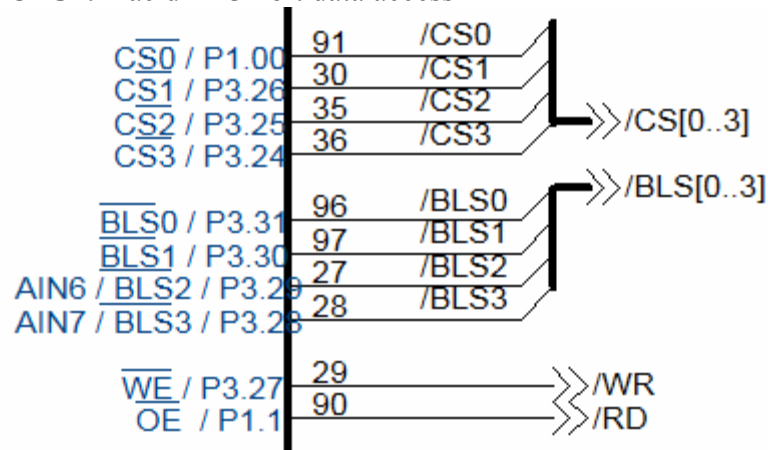
1	8100 0000 - 81FF FFFF	BCFG1
---	-----------------------	-------

Xem lại [bảng 2](#), ta tiến hành các công việc:

1. Chọn cấu hình cho các chân data và địa chỉ: **Port 2&3-GPIO.**
2. Chọn cấu hình cho phép các chân nhập/xuất dữ liệu: $\overline{CS0}, \overline{OE}, BLS[3:0]$.
3. Lựa chọn bank địa chỉ <BCFG0-dùng bank0>
4. Chọn chip 0 < $\overline{CS0} = 0$ > <vùng địa chỉ từ **0x8000 0000-0x80FF FFFF**>
5. $BLS[3:0]=1111$, <ghi cả word>

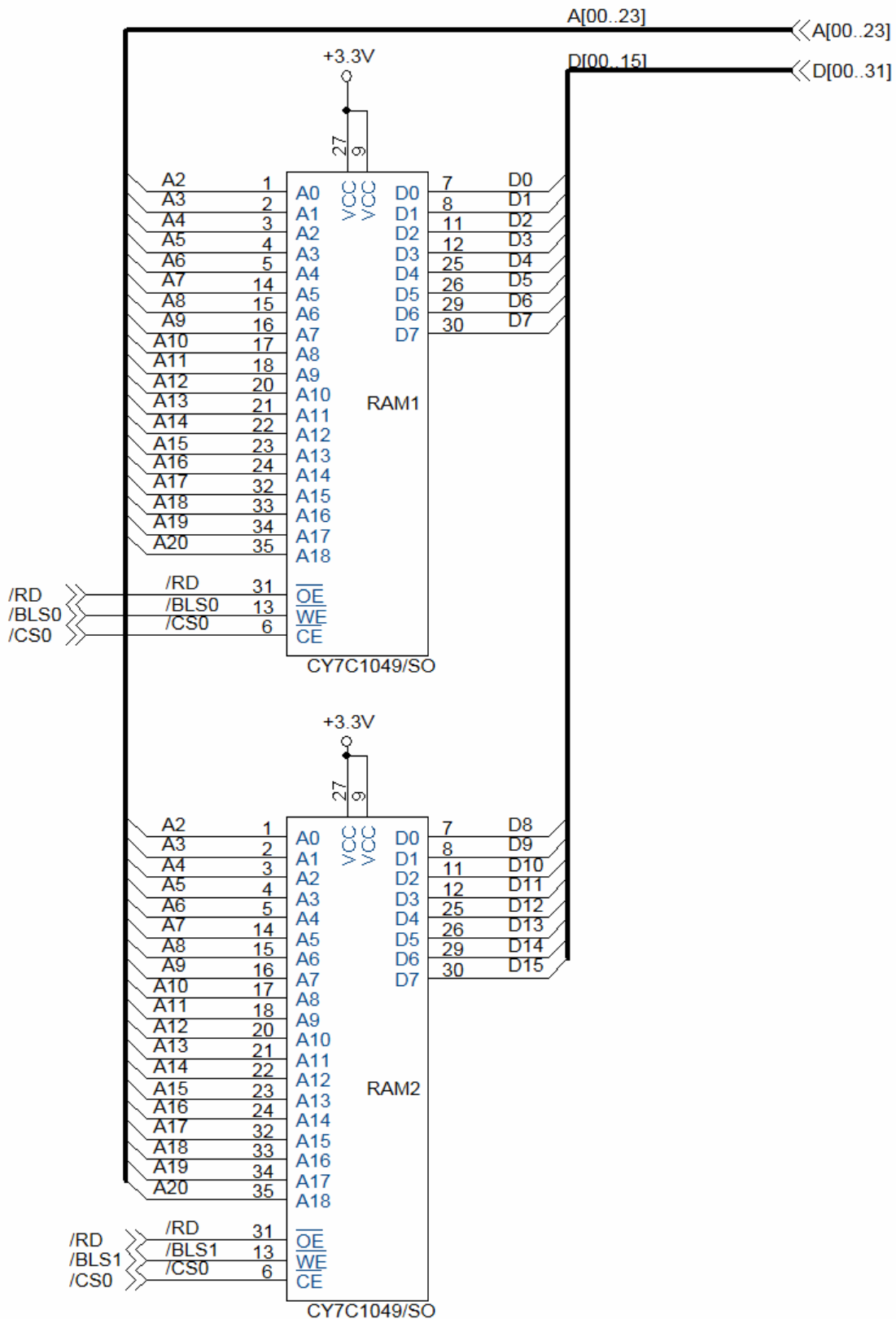
Tiến hành cụ thể:

```
//Cho phép chọn địa chỉ
PINSEL2[27:25]=111;
//Cho phép  $\overline{CS0}$ 
//cho phép  $\overline{OE}$ 
//Cho phép chọn Data:
//Ca word, chọn CS[3:1]
PINSEL2[5:4]=10;
//Lựa chọn Bank BCFG1: mặc định 32 bit data access
```



Hình 8: Lựa chọn cấu hình MCU cho giao tiếp SRAM

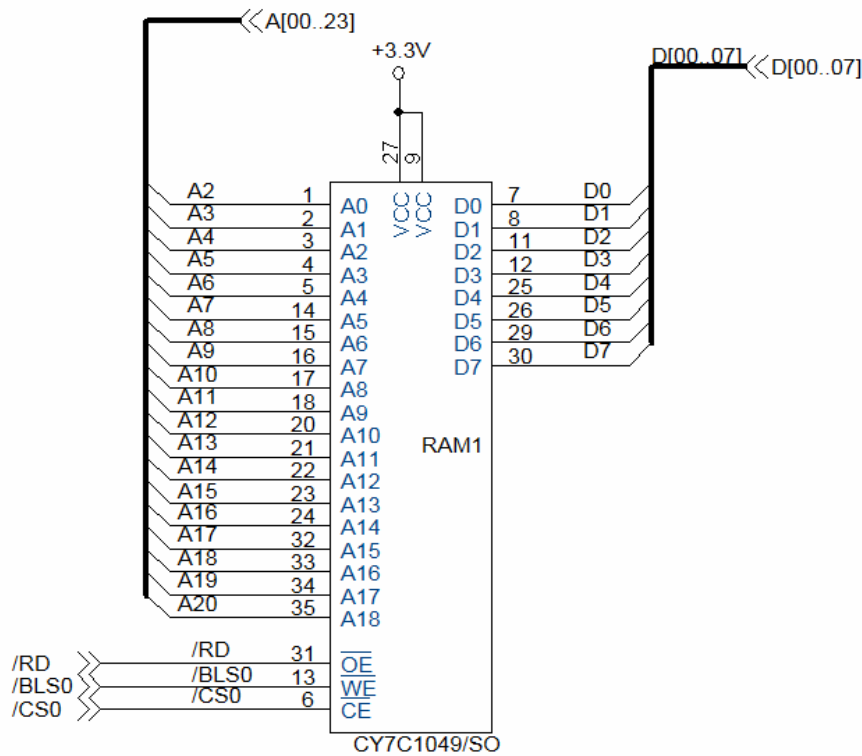
```
//Cho phép chọn địa chỉ
PINSEL2[27:25]=111;
//Cho phép  $\overline{CS0}$ 
//cho phép  $\overline{OE}$ 
//Cho phép chọn Data:
//Half word, chọn CS[1:0]
PINSEL2[5:4]=01;
```



Hình 9: Lựa chọn cấu hình Half Word cho SRAM1 và SRAM2

//Cho phép chặn địa chỉ
PINSEL2[27:25]=111;

//Cho phép $\overline{CS0}$, cho phép \overline{OE} , cho phép chặn Data: byte
PINSEL2[5:4]=00;



Hình 10: Lựa chọn cấu hình Byte cho SRAM1

Đoạn chương trình minh họa cho giao tiếp 8bit SRAM mở rộng, kết quả thu được từ giá trị trong mô phỏng và thực tế:

```

/*****
CHUONG TRINH MINH HOA GIAO TIEP SRAM MO RONG CUA LPC2214
SU DUNG 4 SRAM 512x8 TAO THANH 2MBytes
Tac gia      : BUI TRUNG HIEU, Webmaster: http://www.khvt.com
Huong dan   : ThS HUYNH VAN KIEM, bo mon DKTD, HCMUT
Lien he     : buitrunghieu@khvt.com
Khoi tao    : 20060227
Hoan thanh  : 20060227
Ghi chu:
Test voi mach LPC2214 cua bo mon DKTD, DEE, HCMUT
Thiet ke mach: Nguyen Thanh Phuc, Vitek, HCMC
*****/
#include <lpc22xx.h>
#include <stdio.h>
#include <uart.h>
#include <ram.h>

#define ADDR_TO_P(addr) (*(volatile unsigned char *) (addr))
//Cap phat 10 o nho o bat dau tu dia chi 0x80000000
unsigned char array[10]_at_ 0x80000000;
//dia chi nen la 0x80000000
unsigned long index=0x80000000;

//vao chuong trinh chinh
void main(void)
{
    //Khoi tao cac bien trung gian
    unsigned char temp,i=0;
    //Khoi tao giao tien truyen thong bat dong bo noi tiep
    uart_init();
    printf("\n CHUONG TRINH MINH HOA GIAO TIEP SRAM MO RONG CUA LPC2214\n
SU DUNG 4 SRAM 512x8 TAO THANH 2MBytes\n
Ngươi viet : BUI TRUNG HIEU, Webmaster:
http://www.khvt.com \n
Huong dan : ThS HUYNH VAN KIEM, bo mon DKTD, HCMUT");

```

```

//Khoi tao bus giao tiep RAM 8 bit data
ram_init(8);
//Kiem tra 10 gia tri
while(i<10)
{
    //ghi 10 gia tri tu 0-->9 vao bo nho ngoai SRAM thu 1, du lieu kieu byte
    ADDR_TO_P(index+i)=i;
    //Doc gia tri tu bo nho ngoai SRAM1
    temp=ADDR_TO_P(index+i);
    //In ket qua len man hinh de kiem tra
    printf("\nDia chi %X Gia tri: %d",&ADDR_TO_P(index+i),temp);
    i++;
}
printf("\nQua trinh hoan tat!");
while(1);
}

```

Thư viện “uart.h” và hàm “serial.c” như trong bài UART0 đã nói, hàm khởi tạo “ram.c” được khai báo như sau:

```

void ram_init(unsigned char data_num)
{
    //CHUONG TRINH KHOI TAO SRAM
    /*****
    Chon SRAM:
        +Cho phep /CS0
        +Cho phep /BLS0,/BLS1,/BLS2,/BLS3
        +Cho phep /OE
        +Cho phep chan Data D31:0
    *****/
    switch(data_num)
    {
        //Truong hop chon 32 bit data
        case 32:
            //PINSEL2[5:4]=10;
            PINSEL2|=(1<<5);
            PINSEL2&=~(1<<4);
            //BCFG0[28:29]=10
            BCFG0|=(1<<29);
            BCFG0&=~(1<<28);

            break;

        //Truonghop chon 16 bit data
        case 16:
            //PINSEL2[5:4]=01;
            PINSEL2&=~(1<<5);
            PINSEL2|=(1<<4);
            //BCFG0[29:28]=01
            BCFG0&=~(1<<29);
            BCFG0|=(1<<28);

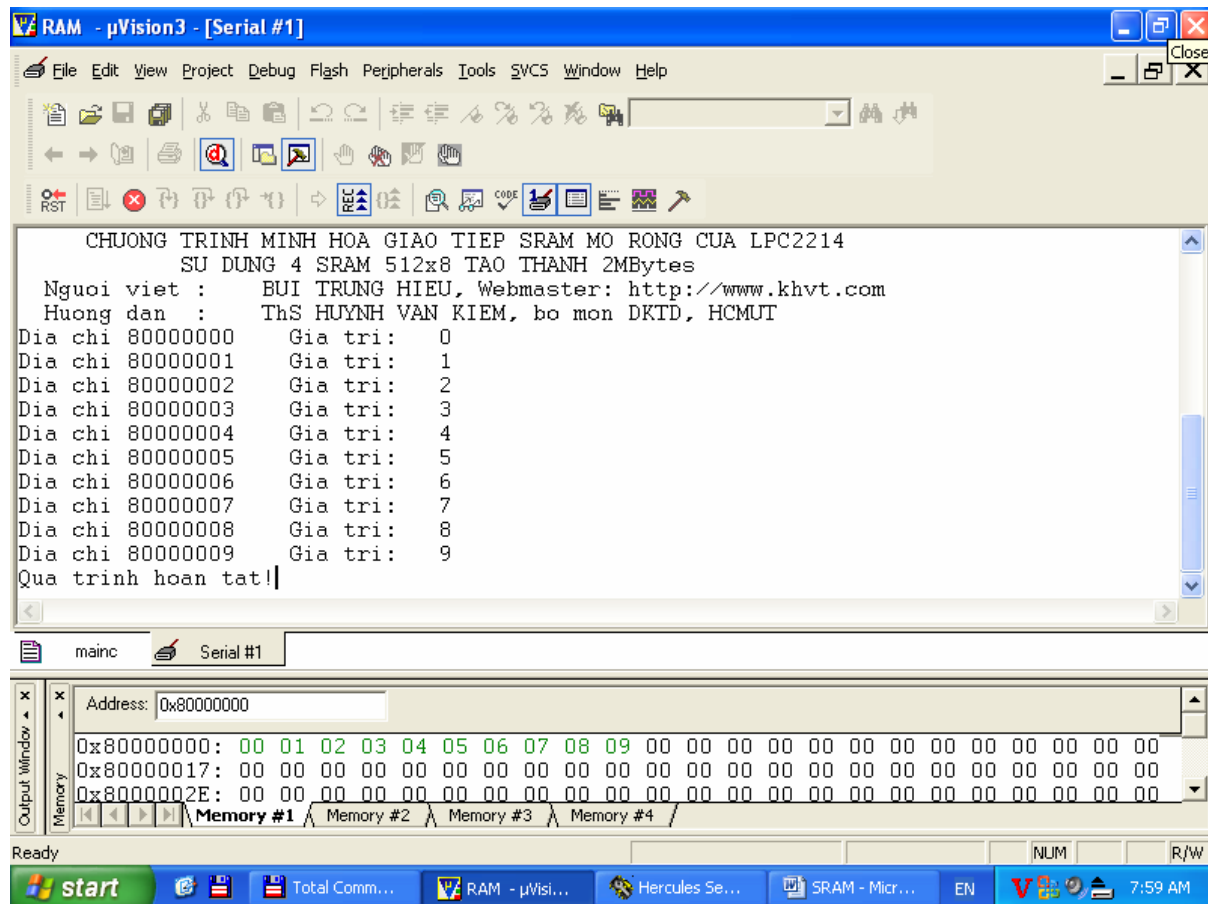
            break;

        //Truong hop chon 8 bit data
        default: //8 bit data
            //PINSEL2[5:4]=00;
            PINSEL2&=~(1<<5);
            PINSEL2&=~(1<<4);
            //BCFG0[28:29]=00
            BCFG0&=~(1<<28);
            BCFG0&=~(1<<29);

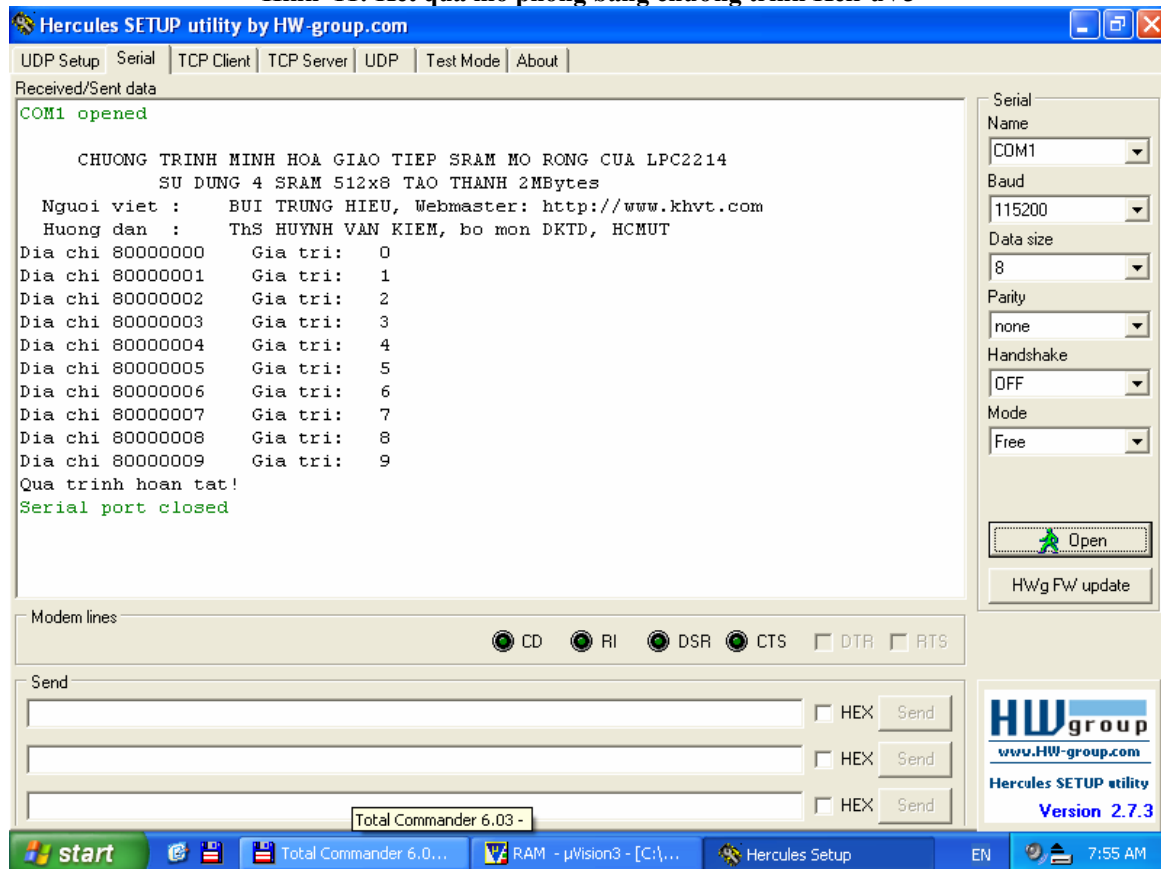
            break;
    }

    //Cho phep chan dia chi A23-A2
    //PINSEL2[27:25]=111;
    PINSEL2|=7<<25;
}

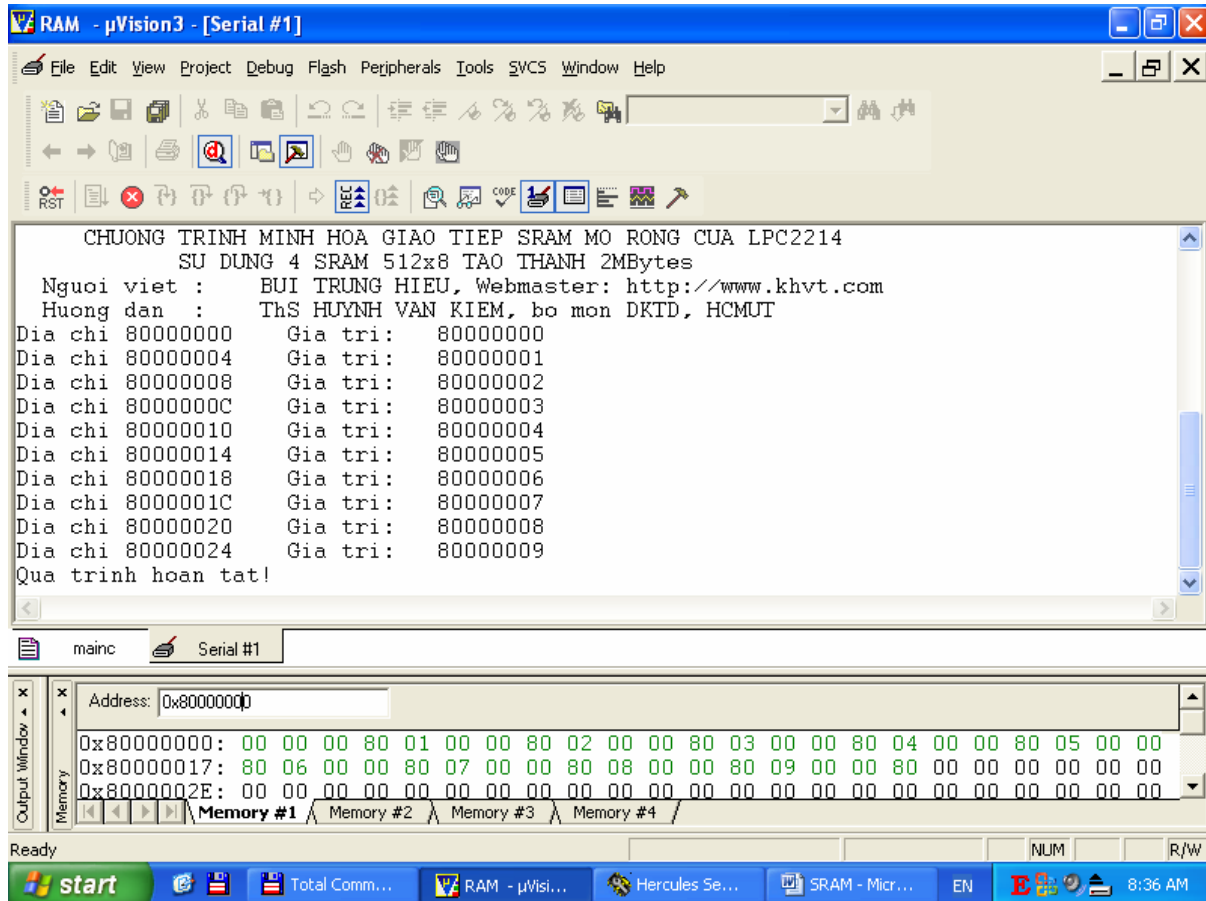
```



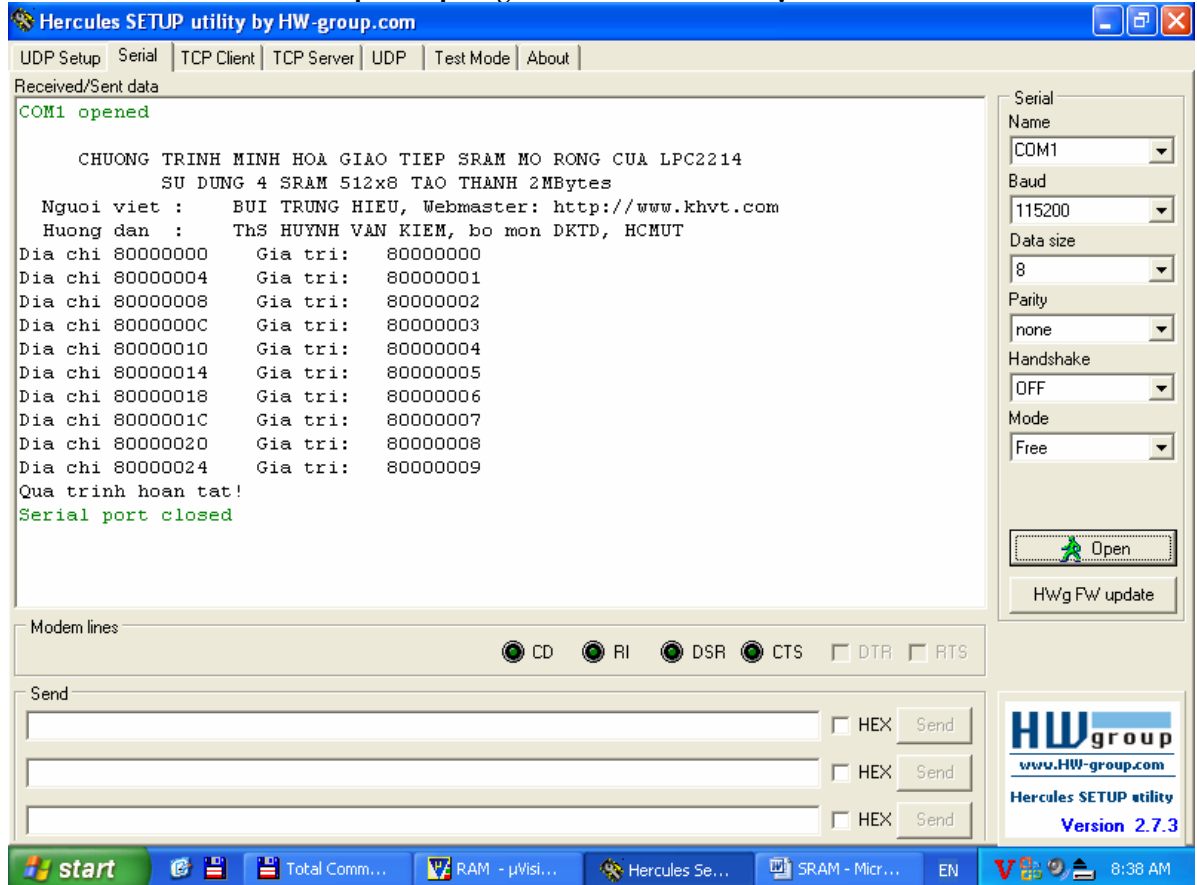
Hình 11: Kết quả mô phỏng bằng chương trình Keil-uV3



Hình 12: Kết quả thật sự với DATA=8 bit



Hình 13: Kết quả mô phỏng với DATA=32 bit-dữ liệu kiểu Little Endian



Hình 14: Kết quả thực tế với 32 bit data

E. Kết luận:

Trong phần này, tôi đã phân tích cách sử dụng SRAM ngoài, với mạch phần cứng cụ thể. Khi hoạt động với nhiều bank, có thể ta cần phải xem xét chi tiết hơn về cách khởi tạo thanh ghi BCFG, thời gian đáp ứng của SRAM... tuy nhiên, với phần cứng đã có, các giới hạn thời gian không xét tới.¹

Quá trình thử mạch thực tế đã cho kết quả rất tốt.

Mọi trao đổi khác <nếu có> về vấn đề này, liên hệ:

✚ Bùi Trung Hiếu
✚ Email : buitrunghieu@khvt.com
✚ Cell : (+84)98.3210.906

¹ Thật ra, tôi đã xét qua các giới hạn này, nhưng không có ảnh hưởng nào từ các giới hạn của SRAM tới hoạt động của MCU.