

标签: [arm](#),[uboot](#),[启动](#),[bootloader](#)

【原创】uboot-2009.01——lowleve_init.S分析



lowleve_init.S分析

```
/*
 * Memory Setup stuff - taken from blob memsetup.S
 *
 * Copyright (C) 1999 2000 2001 Erik Mouw (J.A.K.Mouw@its.tudelft.nl) and
 *      Jan-Derk Bakker (J.D.Bakker@its.tudelft.nl)
 *
 * Modified for the Samsung SMDK2410 by
 * (C) Copyright 2002
 * David Mueller, ELSOFT AG, <d.mueller@elsoft.ch>
 *
 * See file CREDITS for list of people who contributed to this
 * project.
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License as
 * published by the Free Software Foundation; either version 2 of
 * the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston,
 * MA 02111-1307 USA
 */

#include <config.h>
#include <version.h>

/* some parameters for the board */

/*
 *
 * Taken from linux/arch/arm/boot/compressed/head-s3c2410.S
 *
 * Copyright (C) 2002 Samsung Electronics SW.LEE <hitchcar@sec.samsung.com>
 *
 */

@ 我的开发板上的SDRAM为:HY57V561620FTP-H,4Mx16bitx4Banks，两片接成32bit，64MB，接在BANK6上，地址范围为: [0x3000
0000-0x33fffff]
@ （还没搞清楚SDRAM的结构，为什么要分成Banks，改天看下相关的文档研究一下）
```

@ 对于**SDRAM**的初始化是要配置**13**个寄存器，好像多采用循环方式。

```
#define BWSCON 0x48000000
```

@**BWSCON**寄存器地址，控制每个总线

```
/* BWSCON */
```

```
#define DW8 (0x0)
```

```
#define DW16 (0x1)
```

```
#define DW32 (0x2)
```

@ 将来设置位宽的三个宏

```
#define WAIT (0x1<<2)
```

```
#define UBLB (0x1<<3)
```

@ 将来设置**wait**和**ublb**的宏

```
#define B1_BWSCON (DW32)
```

```
#define B2_BWSCON (DW16)
```

```
#define B3_BWSCON (DW16 + WAIT + UBLB)
```

```
#define B4_BWSCON (DW16)
```

```
#define B5_BWSCON (DW16)
```

@**BANK2**、**BANK3**、**BANK4**、**BANK5**为什么配成**16bit**总线宽度，而且**BANK3**还要加上**WAIT**和**UBLB**属性没有搞明白，感觉如果没用的话可以配成默认值的吧？

```
#define B6_BWSCON (DW32)
```

```
#define B7_BWSCON (DW32)
```

@ 我板子的**SDRAM**接在**BANK6**，所以配成**32bit**位宽，**BANK7**按**datasheet**的要求配置的和**BANK6**相同

@ 分别配置每个**BANK**的**BANKCON**寄存器，这里应该根据**SRAM**的手册来配置相应的**BANK**，**BANK0~5**主要是配置一些时序参数

```
/* BANK0CON */
```

```
#define B0_Tacs 0x0 /* 0clk */
```

```
#define B0_Tcos 0x0 /* 0clk */
```

```
#define B0_Tacc 0x7 /* 14clk */
```

```
#define B0_Tcoh 0x0 /* 0clk */
```

```
#define B0_Tah 0x0 /* 0clk */
```

```
#define B0_Tacp 0x0
```

```
#define B0_PMC 0x0 /* normal */
```

```
/* BANK1CON */
```

```
#define B1_Tacs 0x0 /* 0clk */
```

```
#define B1_Tcos 0x0 /* 0clk */
```

```
#define B1_Tacc 0x7 /* 14clk */
```

```
#define B1_Tcoh 0x0 /* 0clk */
```

```
#define B1_Tah 0x0 /* 0clk */
```

```
#define B1_Tacp 0x0
```

```
#define B1_PMC 0x0
```

```
#define B2_Tacs 0x0
```

```
#define B2_Tcos 0x0
```

```
#define B2_Tacc 0x7
```

```
#define B2_Tcoh 0x0
```

```
#define B2_Tah 0x0
```

```
#define B2_Tacp 0x0
```

```
#define B2_PMC 0x0
```

@ 这里BANK3CON不为默认值，是因为在SMDK2410上BANK3上应该接了ROM

@ 根据我自己的板子，我把这里换成了默认值（根据vivi上的默认值修改的），不过要不是默认值的话应该也没有影响的吧？我的板子这里接了DM9000A

```
#define B3_Tacs 0x0 /* 0clk */
#define B3_Tcos 0x3 /* 4clk */
#define B3_Tacc 0x7 /* 14clk */
#define B3_Tcoh 0x1 /* 1clk */
#define B3_Tah 0x0 /* 0clk */
#define B3_Tacp 0x3 /* 6clk */
#define B3_PMC 0x0 /* normal */
```

```
#define B4_Tacs 0x0 /* 0clk */
#define B4_Tcos 0x0 /* 0clk */
#define B4_Tacc 0x7 /* 14clk */
#define B4_Tcoh 0x0 /* 0clk */
#define B4_Tah 0x0 /* 0clk */
#define B4_Tacp 0x0
#define B4_PMC 0x0 /* normal */
```

```
#define B5_Tacs 0x0 /* 0clk */
#define B5_Tcos 0x0 /* 0clk */
#define B5_Tacc 0x7 /* 14clk */
#define B5_Tcoh 0x0 /* 0clk */
#define B5_Tah 0x0 /* 0clk */
#define B5_Tacp 0x0
#define B5_PMC 0x0 /* normal */
```

@ 配置SDRAM的参数，BANK6和BANK7要配置成同样的值，需要查看SDRAM的datasheet

@ 当内存为ROM或SRAM时需要配置所有位，当为SDRAM时只要配置[0~3]

```
#define B6_MT 0x3 /* SDRAM */
```

@ 内存类型，默认为SDRAM

```
#define B6_Trtd 0x1
```

@ RAS到CAS的延时

```
#define B6_SCAN 0x1 /* 9bit */
```

@ 列扫描数，在SDRAM手册上，有一句：C0~C8即为9列

```
#define B7_MT 0x3 /* SDRAM */
#define B7_Trtd 0x1 /* 3clk */
#define B7_SCAN 0x1 /* 9bit */
```

```
/* REFRESH parameter */
```

@ SDRAM刷新参数，这个很重要，移植时要更改

```
#define REFEN 0x1 /* Refresh enable */
```

@ 默认为允许刷新

```
#define TREFMD 0x0 /* CBR(CAS before RAS)/Auto refresh */
```

@ 据说下面三个时间参数多是根据经验来设置

```
#define Trp 0x0 /* 2clk */
```

@ RAS precharge Time

```
#define Trc 0x3 /* 7clk */
```

@ 根据S3C2440的datasheet为：Trc=Trc+Trp，但寄存器中要填入的是Tsrc，这里仿佛应该设为10个时钟才对？

```
#define Tchr  0x2 /* 3clk */

@SDRAM手册上没有找到这个时钟的定义

#define REFCNT  1113 /* period="15".6us, HCLK="60Mhz", (2048+1-15.6*60) */

@SDRAM刷新计数值，这个值比较重要，根据SDRAM的datasheet，（我使用的这款芯片在datasheet写着8192Refresh cycle/64ms，
所以一个刷新周期为64ms/8192=7.8125us,取7.9的话REFCNT的值就为1259）

/*****/

_TEXT_BASE:
.word TEXT_BASE

@ 这里开始lowlevel_init函数了，通过循环连续初始化13个寄存器

.globl lowlevel_init
lowlevel_init:
/* memory control configuration */

/* make r0 relative the current location so that it */

/* reads SMRDATA out of FLASH rather than memory ! */

ldr  r0, =SMRDATA
ldr r1, _TEXT_BASE
@r1放代码段的开始位置

sub r0, r0, r1

//计算SMRDATA的相对地址保存到R0中

//SMRDATA为虚拟地址,而TEXT_BASE为虚拟地址的起始地址

//而现在Uboot的起始地址并不为虚拟地址

//TEXT_BASE为0x33F8 0000,SMRDATA为0x33F8 06C8

//而现在程序运行在起始地址为0x0000 0000的地方

//所以需要计算以0x0000 0000为标准的相对地址

(参考uboot-1.3.0-rc3引导启动学习笔记，这一段确实要好好理解)

ldr r1, =BWSCON /* Bus Width Status Controller */
add  r2, r0, #13*4

@13个寄存器，每个寄存器占4个字节

@ 读取一个值存入寄存器中，然后循环，直到读完13个寄存器

0:
ldr  r3, [r0], #4
str  r3, [r1], #4
cmp  r2, r0
bne  0b

/* everything is fine now */
mov pc, lr

@ 程序返回到cpu_init_crit中

.ltorg
/* the literal pools origin */

SMRDATA:
.word (0+(B1_BWSCON<<4)+(B2_BWSCON<<8)+(B3_BWSCON<<12)+(B4_BWSCON<<16)+(B5_BWSCON<<20)+(B6_BWSCO
N<<24)+(B7_BWSCON<<28))

@ 设置每个BWSCON，注意BANK0由硬件连线决定了。

.word ((B0_Tacs<<13)+(B0_Tcos<<11)+(B0_Tacc<<8)+(B0_Tcoh<<6)+(B0_Tah<<4)+(B0_Tacp<<2)+(B0_PMC))

.word ((B1_Tacs<<13)+(B1_Tcos<<11)+(B1_Tacc<<8)+(B1_Tcoh<<6)+(B1_Tah<<4)+(B1_Tacp<<2)+(B1_PMC))
```

```
.word ((B2_Tacs<<13)+(B2_Tcos<<11)+(B2_Tacc<<8)+(B2_Tcoh<<6)+(B2_Tah<<4)+(B2_Tacp<<2)+(B2_PMC))
.word ((B3_Tacs<<13)+(B3_Tcos<<11)+(B3_Tacc<<8)+(B3_Tcoh<<6)+(B3_Tah<<4)+(B3_Tacp<<2)+(B3_PMC))
.word ((B4_Tacs<<13)+(B4_Tcos<<11)+(B4_Tacc<<8)+(B4_Tcoh<<6)+(B4_Tah<<4)+(B4_Tacp<<2)+(B4_PMC))
.word ((B5_Tacs<<13)+(B5_Tcos<<11)+(B5_Tacc<<8)+(B5_Tcoh<<6)+(B5_Tah<<4)+(B5_Tacp<<2)+(B5_PMC))
```

@ 设置BANKCON0~BANKCON5

```
.word ((B6_MT<<15)+(B6_Trcd<<2)+(B6_SCAN))
.word ((B7_MT<<15)+(B7_Trcd<<2)+(B7_SCAN))
```

@ 设置BANKCON6~BANKCON7

```
.word ((REFEN<<23)+(TREFMD<<22)+(Trp<<20)+(Trc<<18)+(Tchr<<16)+REFCNT)
```

@ 设置REFRESH,在S3C2440中11 ~ 17位是保留的，也即(Tchr<<16)无意义

```
.word 0x32
```

@ 设置BANKSIZE,对于容量可以设置大写，多出来的空内存会被自动检测出来。没有搞明白突发模式操作是什么意思！

```
.word 0x30
```

@ 设置MRSRB6

```
.word 0x30
```

@ 设置MRSRB7

系统分类: [ARM](#) | 用户分类: [嵌入式](#) | 来源: 无分类 | [【推荐给朋友】](#) | [【添加到收藏夹】](#)

该用户于2009/7/11 12:25:23编辑过该文章



顶一下