# Методика определения частоты процессора Байкал-Т1

В документе описана методика измерения частоты процессора Байкал-Т1, требующая минимальные дополнительные программные и аппаратные средства. Поскольку частота работы ядер CPU не может быть измерена непосредственно с помощью внешнего измерительного оборудования, то для определения частоты работы процессора Байкал-Т1 используется метод, основанный на подсчете количества импульсов тактовой частоты ядра CPU за известный интервал времени. В качестве счетчика импульсов тактовой частоты ядра CPU будет использоваться 64-битный счетчик из Глобального Контроллера Прерываний CPU (Global Interrupt Controller - GIC). Для определения частоты ядер CPU потребуется

1. Часы или секундомер - для замера интервала времени
2. Устройство на процессоре Байкал-Т1, например отладочная плата BFK3.1
3. Экспериментатор - человек засекающий время по часам из пункта 1

Экспериментатор в момент времени t1 по часам из пункта 1, снимает показания счётчика Глобального Контроллера Прерываний C1, значение которого увеличивается на единицу с каждым импульсом тактовой частоты ядра CPU, отмеряет произвольный интервал времени и в момент времени t2 по часам из пункта 1 снимает новые показания счётчика Глобального Контроллера Прерываний C2. Частота ядра CPU определяется по формуле

$$F\_CPU = (C2 - C1) / (t2 - t1) \qquad (1)$$

При использовании в формуле (1) времени в секундах, получаемая частота будет в Гц.

Подробно используемый счётчик (GIC CounterLo и GIC CounterHi) описан в разделах 12.5.3.2 и 12.5.3.3 на страницах 591-592 в документе:

MIPS32® P5600 Multiprocessing System Software User’s Manual (Document Number: MD01025 Revision 01.60 April 19, 2016),
доступ к которому можно получить по ссылке:

https://www.mips.com/downloads/p5600-multiprocessing-system-software-users-manual/
MIPS32® P5600 Multiprocessing System Software User’s Manual (Document Number: MD01025 Revision 01.60 April 19, 2016)

Для доступа к состоянию счетчика Глобального Контроллера Прерываний при работе под управлением операционной системы Linux используется программный модуль **clkinfo**. Модуль создает файл /proc/clkinfo, при чтении из которого получается текстовая строка с состоянием 64-битного счетчика Глобального Контроллера Прерываний.

Для сборки и установки модуля **clkinfo** необходимо выполнить следующие действия:

1. В каталог SDK/baikal/src/examples распаковать clkinfo.tgz или скопировать в файл clkinfo_main.c программный текст модуля из Приложения 1 настоящего документа и создать makefile следующего содержания:

```
obj-m  += clkinfo.o
clkinfo-objs := clkinfo_main.o
PWD   := $(shell pwd)

all:
	$(MAKE) -C $(KDIR) SUBDIRS=$(PWD) modules
clean:
	$(MAKE) -C $(KDIR) SUBDIRS=$(PWD) clean
```

2. Произвести сборку модуля командой:

```
make ARCH=mips CROSS_COMPILE=$(pwd)/../../../usr/x-tools/mipsel-unknown-linux-gnu/bin/mipsel-unknown-linux-gnu- KDIR=../../kernel
```

3. Установить полученный модуль clkinfo.ko командой

```
insmod clkinfo.ko
```

Для проверки работоспособности модуля исполнить команду

```
cat /proc/clkinfo
```

на экране появится строка с состоянием счетчика Глобального Контроллера Прерываний, например

```
cat /proc/clkinfo
clk 3288428330498
```

Определение частоты ядра CPU можно произвести далее по процедуре, описанной в данном документе ранее.

Для определения частоты ядра CPU можно использовать также следующую последовательность команд

```
cat /proc/clkinfo ; sleep 60 ; cat /proc/clkinfo
```

Исполнение этой последовательности команд выведет на экран состояние счетчика Глобального Контроллера Прерываний на момент запуска последовательности команд и состояние счетчика Глобального Контроллера Прерываний через 60 секунд (sleep 60), например

```
cat /proc/clkinfo ; sleep 60 ; cat /proc/clkinfo
clk 3775040436716
clk 3847049482604
```

Для расчета частоты ядра CPU используем формулу (1):

F_CPU = (3847049482604 - 3775040436716) / 60 = 72009045888 / 60 = 1 200 150 764,6 Гц

Время задержки 60 секунд можно контролировать по внешним часам. Вместо задержки в 60 секунд можно использовать и другую величину задержки.

## **Приложение 1.** Исходный код модуля **clkinfo**

```
/*
 * Enable user-mode MIPS counter access.
 */
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/smp.h>
#include <linux/miscdevice.h>
#include <linux/slab.h>
#include <linux/uaccess.h>
#include <linux/fs.h>

#include <linux/proc_fs.h>
#include <linux/seq_file.h>

static u64
getTicks(void) {
	u32	tmp, hi, lo;
	u64	ticks;
	do{
		hi = (*((volatile u32 *)(0xBBDC0000 + 0x14)));
		lo = (*((volatile u32 *)(0xBBDC0000 + 0x10)));
		tmp = (*((volatile u32 *)(0xBBDC0000 + 0x14)));
	} while(hi != tmp);
	ticks = hi;
	ticks = (ticks << 32) + lo;
	return ticks;
}

static char temp[512];

static int clk_print(struct seq_file *m, void* prm)
{
	snprintf(temp, sizeof(temp),
		"clk %lld\n",
		getTicks()
		);
	seq_puts(m, temp);
	return 0;
}

static ssize_t clkinfo_proc_write(struct file *file, const char __user *buffer,
                                  size_t count, loff_t *ppos)
{
	return count;
}
```

```
static int clkinfo_proc_open(struct inode *inode, struct file *file)
{
	return single_open(file, clk_print, NULL);
}

static const struct file_operations clkinfo_proc_fops = {
	.owner		= THIS_MODULE,
	.open		= clkinfo_proc_open,
	.read		= seq_read,
	.llseek		= seq_lseek,
	.release	= single_release,
	.write		= clkinfo_proc_write,
};

static struct proc_dir_entry *clkinfo_proc;


static int __init
init(void)
{

	clkinfo_proc = proc_create("clkinfo", 0, NULL, &clkinfo_proc_fops);

	if (clkinfo_proc == NULL) {
		printk(KERN_ERR "failed /proc/clkinfo\n");
		return 1;
	}

	return 0;
}

static void __exit
fini(void)

{
	remove_proc_entry("clkinfo", NULL /* parent dir */);
}

MODULE_DESCRIPTION("Enables user-mode access to MIPS counters");
MODULE_LICENSE("GPL");
module_init(init);
module_exit(fini);
```