

Имитационное моделирование в исследовании и разработке информационных систем

Лекция 2

.Наблюдение за работой системы

- . Аппаратные наблюдатели
- . Программные наблюдатели
- . Анализ результатов наблюдения

Отладка программы (или управляющей системы)

- Наблюдение за поведением системы
- Проверка соблюдения заданных свойств поведения
- Воздействие на систему
- Программные и аппаратные наблюдатели

Аппаратные наблюдатели

- Встроенные в аппаратуру выч. сист.
 - Отладочные регистры процессора
 - JTAG
- Универсальные внешние
 - Электронный (цифровой) осциллограф
 - Логический анализатор
- Специализированные

Отладочные регистры на x86

- 6 регистров: DR0-DR3, DR6, DR7
- Точки останова (4 шт.):
 - по чтению;
 - по записи и чтению;
 - по выполнению;
 - по обращению к порту ввода-вывода.
- Нельзя задать условие по диапазону адресов

Встроенные отладочные средства (JTAG – Joint Test Action Group)

- Набор специальных блоков в составе ИС (существенно зависит от типа ИС)
- Управление этими блоками по интерфейсу JTAG
- Возможность разрывать связи между блоками ИС, устанавливать заданные значения на разорванных связях
- Управление микропроцессором в целях отладки: точки останова, просмотр и запись регистров

Универсальные наблюдатели

.Индикаторы

.Электронный осциллограф

- Непрерывные и дискретные сигналы
- Обычно 1-4 канала
- Непрерывная и ждущая развертка
- Задержка запуска развертки

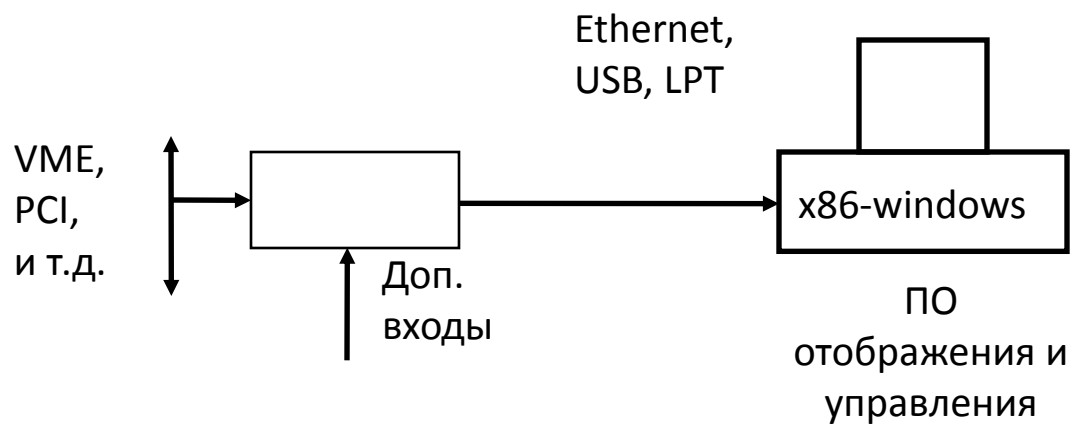
.Логический анализатор

- Цифровой сигнал
- Десятки-сотни каналов
- Задание условий запуска и окончания регистрации

Шинный интерфейс

- Объединяет несколько устройств (абонентов) (десятки-сотни)
- Один набор сигнальных линий для обменов между всеми абонентами (*есть исключения для служебных линий*)
- В каждый момент времени возможна передача только одним абонентом
- Процедура арбитража для избежания конфликтов
- Стандартизованы: протоколы, электрические и механические характеристики

Анализаторы шины

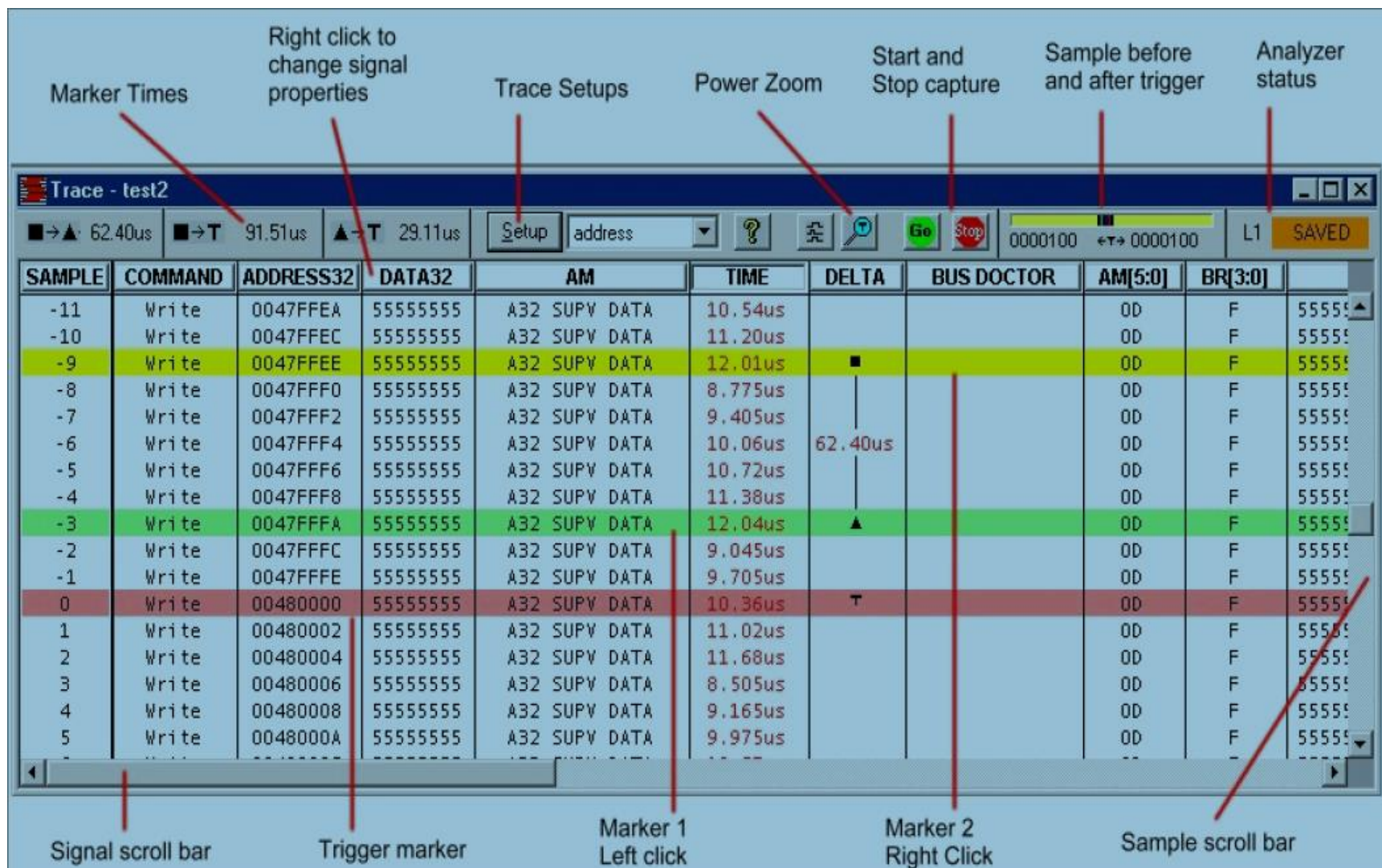


- Гранит-ВТ
- Silicon Control Inc
- Curtiss-Wright Electronics systems (быв VMETRO)
- LeCroy Inc
- Tektronix Inc

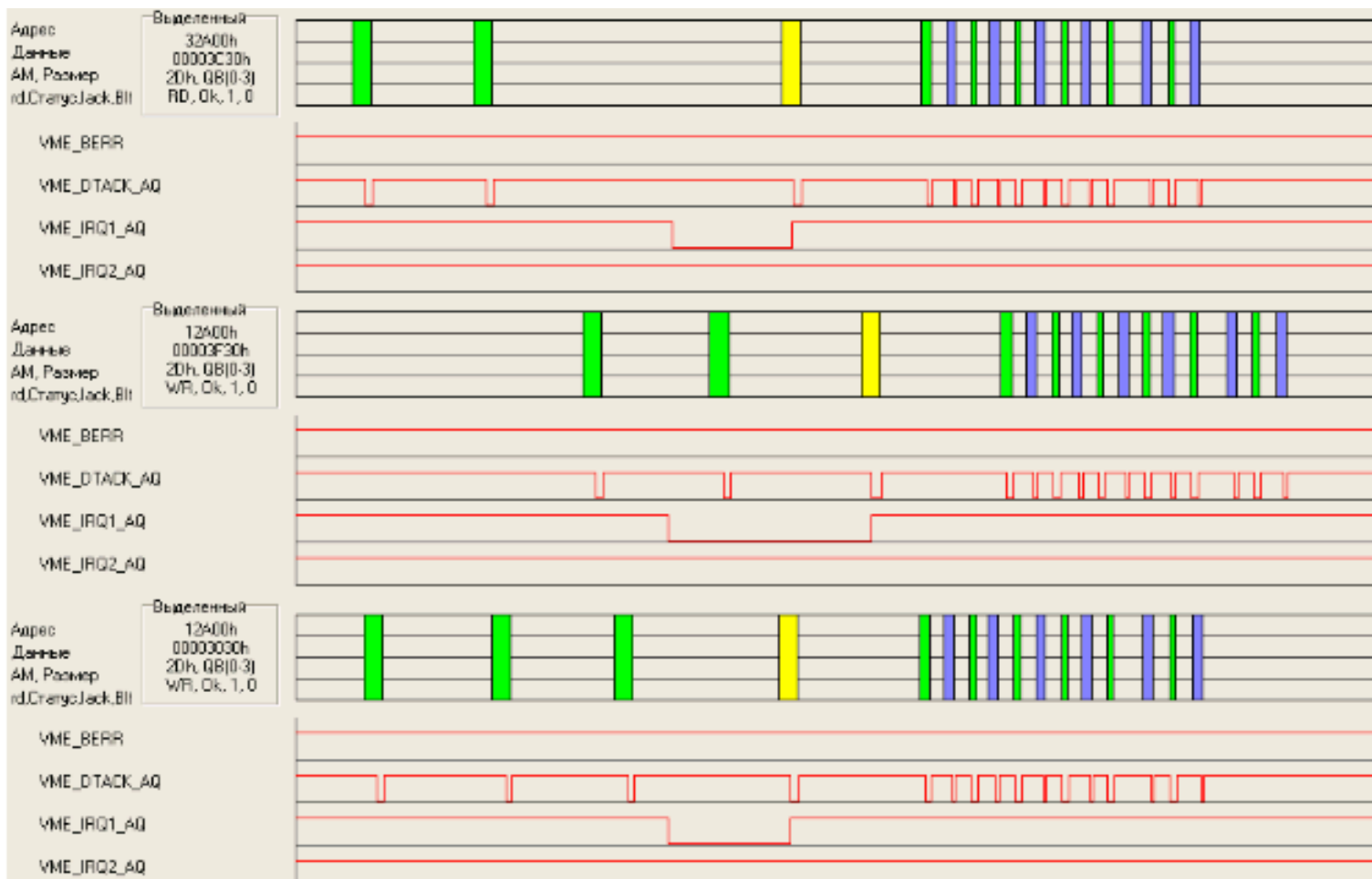
Настройки анализатора шины

- Условия начала и окончания регистрации
- Что регистрировать (фильтр)
- Как отображать

Вид 1: таблица обменов



Вид 2: временная диаграмма



Выбор событий для регистрации

- Регистрация обменов
- Регистрация сигналов
 - Выбор отдельных сигналов
- Условие начала регистрации
- Условие записи сигнала (фильтр)

Выбор событий для регистрации (2)

Trace Setup

Setup
Name:
Description:

Setup Wizard 

Events

EVENT	COMMAND	ADDRESS32	=<>	DATA32	=<>	AM	AD[63:0]	AM[5:0]	ANON
E1:	All	XX480000	Any	XXXXXXXX	Any	All	XXXXXXXX XX480000	XX	Non
E2:	All	XXXXXXXX	Any	XXXXXXXX	Any	All	XXXXXXXX XXXXXXXX	XX	Non
E3:	All	XXXXXXXX	Any	XXXXXXXX	Any	All	XXXXXXXX XXXXXXXX	XX	Non
E4:	All	XXXXXXXX	Any	XXXXXXXX	Any	All	XXXXXXXX XXXXXXXX	XX	Non

Trace Control

Sample Clock: Trigger Position: Power Zoom Sample Clock:

L1: IF OCCURS THEN
STORE ELSE

L2: IF OCCURS THEN
STORE ELSE

L3: IF OCCURS THEN
STORE ELSE

Выбор событий (3)

- Элементарное событие
 - Условие на состояние групп сигналов шины (=, <, >, от .. до)
 - Ограниченное число событий (8, 32)
- Выражение над событиями (И, ИЛИ, НЕ)
- Конечный автомат (8 состояний)
 - Условие перехода в новое состояние
 - Что записывать, находясь в данном состоянии
 - Возбуждать ли триггер

Анализатор: интерфейс пользователя

- Прокрутка
- Измерение интервалов времени
- Фильтр по группам сигналов шины
- Поиск по группам сигналов шины
- Сохранение трассы
- Сравнение трасс
- Программное управление (API)

Анализ производительности

- Загруженность
 - BusBusy, BusIdle, DataPhase, WaitState
- Пропускная способность
 - Все обмены, или по приоритетам
 - По диапазонам адресов
- Запаздывание
 - Master от выставления адреса до первого слова данных
 - Target от запроса первого слова данных до

Имитация абонента шины

- Инициатор (master)
 - Адрес, длина
 - Режим (пословный, пакетный)
 - Запись, чтение, запись с проверкой
- Подчиненное устройство (target)
 - Диапазон адресов
 - Число тактов ожидания
 - Ответ (OK, retry, fail)

Имитация протокола шины

Stimulus

Go Stop IDLE

Stimulus

STIMULUS	COMMAND	ADDRESS32	=<>	DATA32	=<>	AM	AD[63:0]
S1:	All	XXXXXXXX	Any	XXXXXXXX	Any	All	XXXXXXXX XXXXXXXX
S2:	All	XXXXXXXX	Any	XXXXXXXX	Any	All	XXXXXXXX XXXXXXXX
S3:	All	XXXXXXXX	Any	XXXXXXXX	Any	All	XXXXXXXX XXXXXXXX
S4:	All	XXXXXXXX	Any	XXXXXXXX	Any	All	XXXXXXXX XXXXXXXX
S5:	All	XXXXXXXX	Any	XXXXXXXX	Any	All	XXXXXXXX XXXXXXXX

Events

EVENT	COMMAND	ADDRESS32	=<>	DATA32	=<>	AM	AD[63:0]
E1:	All	XXXXXXXX	Any	XXXXXXXX	Any	All	XXXXXXXX XXXXXXXX
E2:	All	XXXXXXXX	Any	XXXXXXXX	Any	All	XXXXXXXX XXXXXXXX

Control

SL1: IF OCCURS THEN
STIMULUS ELSE

SL2: IF OCCURS THEN
STIMULUS ELSE

Stimulus Control

Event Conditions

Программные наблюдатели:

уровень операционной

Трассировка системных системы ВЫЗОВОВ

% strace /bin/cat

// Динамическая загрузка

```
open("/lib/i386-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0000\226\1\0004\0\0\0"...
512) = 512
fstat64(3, {st_mode=S_IFREG|0755, st_size=1734120, ...}) = 0
mmap2(NULL, 1743580, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0xb75a0000
mmap2(0xb7744000, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a4) = 0xb7744000
mmap2(0xb7747000, 10972, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0xb7747000
close(3) = 0
```

Вывод strace(2)

```
. read(0, "fkjdfkjfd\n", 32768)
= 10
write(1, "fkjdfkjfd\n", 10)
= 10
read(0, "sdcdscdsv\n", 32768)
= 10
write(1, "sdcdscdsv\n", 10)
= 10
read(0, "fbfb\n", 32768)
= 5
write(1, "fbfb\n", 5)
```

Трассировка библиотечных

Функций

• % ltrace /bin/cat 2> ltrace.out

```
__libc_start_main(0x8049280, 1, 0xbfb66534, 0x80500d0, 0x8050140
<unfinished ...>
getpagesize() = 4096
strchr("/bin/cat", '/') = "/cat"
setlocale(6, "") =
"en_US.UTF-8"
bindtextdomain("coreutils", "/usr/share/locale") =
"/usr/share/locale"
textdomain("coreutils") =
"coreutils"
__cxa_atexit(0x804a2f0, 0, 0, 0xb7769ff4, 0) = 0
getopt_long(1, 0xbfb66534, "benstuvAET", 0x08050840, NULL) = -1
__fxstat64(3, 1, 0xbfb6641c) = 0
__fxstat64(3, 0, 0xbfb6641c) = 0
```

Трассировка библиотечных функций (2)

<code>posix_fadvise64(0, 0, 0, 0, 0)</code>	<code>= 0</code>
<code>malloc(36863)</code>	<code>= 0x097e0890</code>
<code>read(0, "qwerty\n", 32768)</code>	<code>= 7</code>
<code>write(1, "qwerty\n", 7)</code>	<code>= 7</code>
<code>read(0, "", 32768)</code>	<code>= 0</code>
<code>free(0x097e0890)</code>	<code>= <void></code>
<code>close(0)</code>	<code>= 0</code>
<code>exit(0 <unfinished ...></code>	

Время в операционной системе

```
#include <time.h>
```

```
struct timespec {
```

```
time_t tv_sec;
```

```
int tv_nsec; // нано
```

```
};
```

```
struct tm {}
```

```
time()
```

```
sleep()
```

Измерение времени в ОС (2)

- `/usr/bin/time myfile`
 - real
 - user
 - sys
- Команда `rdtsc` в x86 – число тактов ЦП с момента запуска

Профилирование программ: gprof

% gcc -pg a.c

% a.out *(создаётся gmon.out)*

% gprof

Выборка 100 раз в секунду

- Число вызовов функций
- Время, проведённое в каждой функции
- Граф вызовов

valgrind

- `% valgrind myfile`
- `% valgrind -- tool=cachegrind myfile`
- Проверка некоторых условий корректности работы с указателями и `malloc/free`
- Анализ работы гипотетической кэш-памяти
- Построение графа вызовов функций

Возможность добавления инструментов

Лабораторная работа 1

- Измерение времени выполнения программы (файл b.c)
- Используйте
 - команду `time`
 - `rdtsc` (как вставлять `asm` в Си?)
- Повторите замер несколько раз
- Как уменьшить разброс?
- Поменяйте порядок обхода массива. Почему изменилось время?

Программа для замера времени выполнения

```
#include <stdio.h>
#define SIZE 10000
float a[SIZE][SIZE];

void main( int argc, char **argv )
{
    float b; int i,k;
    b=0;
    for( i = 0; i < SIZE; i++ )
        for( k = 0; k < SIZE; k++ )
            b = b + a[i][k];
}
```

Варианты задания

- Замер любых других временных характеристик вычислительной системы:
- Объяснение, как проводилось измерение; меры по снижению погрешности;
- Результаты;
- Изменение настроек, что получилось и почему?