

Язык TCL для проектирования на Xilinx

Часть 1 Основы TCL

Владимир Викулин, инженер по применению Xilinx

16.12.2021 г.

План вебинара

- ◆ Введение: TCL –Что это такое
- ◆ Основные документы и ссылки
- ◆ TCL в Vivado
- ◆ Основы TCL
- ◆ Обработка строк
- ◆ Процедуры и функции, области видимости
- ◆ Списки и ассоциативные массивы
- ◆ Основные конструкции языка
- ◆ Пример реализации проекта через tcl скрипт
- ◆ Заключение и ответы на вопросы



TCL – что это такое?

TCL – скриптовый язык программирования, созданный в 1988 г. [Джоном Остерхаутом](#).

Идеология TCL: Все объекты – это строки

Основная идея – интерпретация строк

Преимущества:

- возможность использования для интеграции различных приложений в единую систему
- чрезвычайная гибкость
- удобство работы со строками, включая регулярные выражения
- интерактивность и отличная интеграция с ОС

TCL – что это такое? (2)

Недостатки:

- низкое быстродействие
- необычный синтаксис
- отсутствие каких-бы то ни было защит

TCL де-факто является языком автоматизации и интеграции во всех ведущих программных пакетах разработки микросхем (ПЛИС и ASIC). Именно поэтому мы его изучаем!

Документация по TCL

- ◆ Статья в Википедии <https://ru.wikipedia.org/wiki/Tcl>
- ◆ Книга "Практическое программирование на Tcl и Tk" Брент Б. Уэлш, Кен Джонс, Джеффри Хоббс ISBN: 5-8459-0661-X
- ◆ Tcl Developer Xchange <https://www.tcl.tk/>
- ◆ Онлайн-учебник <https://www.tcl.tk/man/tcl8.5/tutorial/tcltutorial.html>

Документация Xilinx по TCL

- ◆ UG975 vivado quick reference guide
- ◆ UG835 - Vivado Design Suite Tcl Command Reference Guide
- ◆ UG894 - Vivado Design Suite User Guide: Using Tcl Scripting
- ◆ UG903 - Vivado Design Suite User Guide: Using Constraints

TCL в Vivado

Vivado всецело основана на TCL

- TCL является основой как интерфейса с пользователем, так и внутреннего представления всех объектов.
- Каждое действие IDE транслируется в TCL и протоколируется
- Имеется TCL-консоль встроенная в IDE и отдельная TCL-shell (vivado - mode tcl)
Пример: vivado -mode batch -source script.tcl -tclargs "FPGA=115-2"
- Любой проект может быть реализован в командном режиме с помощью скриптов на TCL
- Несколько примеров использования TCL в Vivado IDE

ОСНОВЫ TCL

Команда	Описание	Примеры
Вызов TCL shell	Вызов интерпретатора	<code>vivado.bat -mode tcl</code>
Команда <code>set</code>	Оператор присваивания	<code>set hi hello</code> <code>set hi {hello}</code>
Команда <code>Puts</code> и программа Hello World на TCL	Вывод на консоль	<code>puts {Hello world}</code> <code>puts "Hello world"</code> <code>puts "\$hi world"</code> <code>puts \$hi</code> <- это одно <code>puts \${hi}</code> <- и то же
Команда <code>Source</code>	Вызов скрипта	<code>source [-notrace] <name>.tcl</code>

Обработка строк – основные правила

Пример	Некоторые функции
Это не строка	string compare [-nocase]
А_это_строка	string equal
{Это тоже строка}	string length
“И \$это строка”	string ...
[Это_особая_строка]	format
Ограничители строк: “” {}	subst
	eval

Обработка строк - регулярные выражения

Синтаксис:	<code>regexp opSwitches patterns searchString fullMatch subMatch1 ... subMatchn</code>
<code>opSwitches</code>	<code>-nocase -indices -line -start index --</code>
<code>patterns</code>	шаблон
<code>searchString</code>	строка в которой ищем

Подробнее: <https://www.tcl.tk/man/tcl8.5/tutorial/Tcl20.html>

Процедуры и функции, области видимости

Процедуры и функции, области видимости

Объявление	Вызов
<pre># Процедура с одним параметром proc ftc {f} { return [expr \$f * 9 / 5 + 32] }</pre>	<pre>[ftc 200] [ftc \$curtemp]</pre>
<pre>Процедура без параметров proc const5 {} { return 5 }</pre>	<pre>[const5]</pre>
<pre>Глобальные переменные set opA 2 set opB 5 proc gsum {} { global opA global opB return [expr \$opA + \$opB] }</pre>	<pre>[gsum]</pre>

Управление видимостью	пример
global	global var
upvar	<pre>proc swap_upvar {a b} { upvar \$a op1 upvar \$b op2 set tmp \$op1 set op1 \$op2 set op2 \$tmp }</pre>

Списки

Списки совместно с foreach

```
set i 1
foreach value {1 2 3 4 5} {
    set i [expr $i*$value]
}
puts $i
>120
```

Списки (2)

<pre># Списки set myColors {red green blue} ;# список set myColors "red green blue" ;# то же самое # Можно задать через функции set myColors [list red green blue] set myColors [split "red_green_blue" _]</pre>	Определение списка
index, append, leppend, length, lsort	Основные функции обработки списков
<pre># Пример puts [index \$myColors 1] puts [length \$myColors]</pre>	Результат: green 3
Пример	Скрипт tcl_lists.tcl

Массивы

Массивы в TCL не простые, а ассоциативные, т.е. это пара ключ – значение

Из комбинации ассоциативных массивов и списков можно создавать сложные структуры, вплоть до баз данных.

Пример: скрипт `tcl_arrays.tcl`

Основные управляющие конструкции

Управляющие конструкции	Пример
if	<pre>if expr1 ?then? body1 elseif expr2 ?then? body2 elseif ... ?else? ?bodyN?</pre>
?: (динозавр)	<pre>set age 32 set adult [expr \$age >= 18 ? true : false] puts "Adult: \$adult"</pre>
switch	
while	<pre>while {cond} { # loop body }</pre>
for	<pre>for {init_expr} {end_expr} {inc_expr} { # loop body }</pre>
foreach	<pre>foreach {loop_vars} list { # loop body }</pre>

Пример switch

```
set flag [lshift args] # чтение командной строки
```

```
switch -exact -- $flag {
```

```
  -p -
```

```
  -ports {
```

```
    set ports [lshift args]
```

```
  }
```

```
  -v -
```

```
  -verbose {
```

```
    set verbose 1
```

```
  }
```

```
  -h -
```

```
  -help {
```

```
    incr help
```

```
  }
```

```
  default {
```

```
    ....
```

```
  }
```

```
}
```


Пример цикла while

```
set x 1
```

```
while {$x < 5} {  
    puts "x is $x "  
    set x [expr {$x + 1}]  
}
```

Пример цикла for

```
for {set i 0} {$i < 10} {incr i} {  
    puts "I inside first loop: $i"  
}
```

Пример цикла foreach

```
set lst "a b c"
```

```
set i 0
```

```
foreach listitem $lst {
```

```
    puts "$ listitem is item number $i in list lst"
```

```
    incr i
```

```
}
```

Пример проекта TCL в Vivado

Пример создания tcl-проекта из обычного проекта Vivado

- Зачем? – Например, для включения в систему автоматической сборки
- Как? – С помощью копирования команд из TCL Console и их незначительной модификации
- Пример? – arty_tcl

Спасибо за внимание !

Ваши вопросы



Наши ответы



**МАКРО
ГРУПП**

Официальный дилер Xilinx

Контакты

Тел.: 8 (800) 333-06-05

email: SALES@MACROGROUP.RU

Продукция Xilinx и техподдержка: fpga@macrogroup.ru

Силовая электроника: power@macrogroup.ru

Олег Болихов – руководитель направления “Цифровая электроника”

Дмитрий Хорьков – руководитель направления Xilinx

Владимир Викулин, Дмитрий Шадрин – техподдержка Xilinx

Ссылки

Страница TrenZ Electronic:

<https://www.trenz-electronic.de/en/>

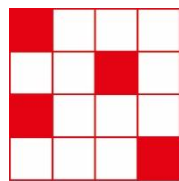
TrenZ Electronic EDDP: <https://wiki.trenz-electronic.de/display/PD/EDDP+Resources>



Совместная программа обучения



**МАКРО
ГРУПП**



PLC2

- ◆ Технологии Xilinx сложны и разнообразны, поэтому квалификация разработчиков имеет решающее значение для успешного выполнения проектов
- ◆ Обучение по программам и стандартам лидера европейского обучения PLC2 – авторизованного партнера тренинг-партнера Xilinx (XTP)
- ◆ На русском языке
- ◆ Сертификат PLC2 + Макро

Xilinx – полезные ссылки

- ◆ Сайт Xilinx: <https://www.xilinx.com/>
- ◆ Сайт Developer: <https://developer.xilinx.com/>
- ◆ Форум: <https://forums.xilinx.com/>
- ◆ Обучение: <https://xilinxprod-catalog.netexam.com/>
- ◆ Репозиторий: <https://github.com/Xilinx>
- ◆ Отладочные платы и платформы:
<https://www.xilinx.com/products/boards-and-kits/see-all-evaluation-boards.html>

Xilinx – материалы с сайта

Справочные и методологические материалы на сайте Xilinx

- ◆ DocNav – все в одном месте на вашем компьютере
- ◆ Selection guides – руководства по выбору
- ◆ User guides – руководства по применению
- ◆ UFSM – методология проектирования
- ◆ AR – ответы на вопросы

Xilinx – как получить техподдержку

Техподдержка

- ◆ Сначала посмотреть на форуме Xilinx
- ◆ Поискать на ресурсах Xilinx:
<https://www.xilinx.com/support.html>
- ◆ Обратиться в Macro: fpga@macrogroup.ru
- ◆ Открыть service request:
https://service.xilinx.com/sservice_prod/start.swe

Xilinx – на чем разрабатывать и отлаживать свои проекты

Отладочные платы и платформы

Подберем отладку под ваши задачи

- <https://www.macrogroup.ru/catalog/partgroup/378>
- <https://www.xilinx.com/products/boards-and-kits.html>

