ВВЕДЕНИЕ В АРХИТЕКТУРУ зла синтеза Цифровых зла синтеза цифро

школа синтеза цифровых схем Школа синтеза цифро

**RISC-V** 

Школа синтеза цифровых схем

ЯЗЫК АССЕМБЛЕРА

Школа синтеза цифровых схем Та *Синтеза* <u>Чифровых</u> схем

Школа синтеза цифров Школа синтеза цифрову **TPW TAPTHEPCTBE** 

Занятие №17

ла си**15** февраля 2025 Чифровых схем

Школа синтеза цифровых схем

Школа синтеза <u>Чифровінх схем</u>

Школа синтеза цифров

ла синтеза <u>Чифровых</u> схем эла синтеза Цифровых схем эла синтеза Цифровых схем зла синтеза цифро

Школа синтеза цифровых схем Школа синтеза цифро Школа синтеза чиф ола синтеза цифровых схем Никита Поляков

YADRO, ведущий инженер Школа синтеза, инженер-ведущий

.005 - 2011: М. Т. ... 2008 - 2020: МЦСТ. Эльбрус 2020 - н.в.: Синтакор. RISC-V Антеза цифровых схем

та синтеза <u>Чифровых схем</u>

Школа синтеза цифровую схем Школа синтеза цифров

Школа синтеза цифров

Школа синтеза цифровых схем 7a синтеза Цифровых схем

Процесс вычисления или выполнения программы может быть представлены на нескольких уровнях абстракции

— Каждому уровню абстракции соответствует средство проектирования теза чифровых схем

Уровень абстракции

Алгоритм

Архитектура набора команд

Программа на машинном коде

Чифровых схем

Языки высокого уровня (С/С++)

Язык ассемблера, который понятен человеку

Двоичный код, который "понятен" аппаратуре

a = b + c;

add x4, x2, x3

0x00310233

Микроархитектура

ем ф. Схем

Блок-схемы и

Школа синтеза Чифровых схем

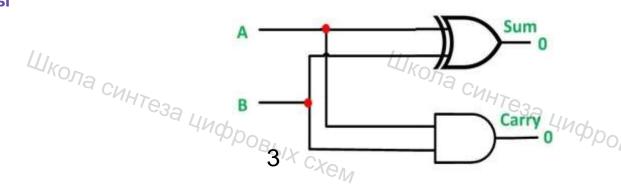
языки описания апларатуры (Verilog, VHDL)

Электрические схемы

assign x4 = x2 + x3;

ола синтеза цифр<sub>с</sub>

Школа синтеза цифро



программа

язык ассемылера
КОМАНДЫКПРОЦЕССОРА

КОМАНДЫКПРОЦЕССОРА

КОМАНДЫКПРОЦЕССОРА

КОМАНДЫКПРОЦЕССОРА

• Работа процессора (Central Processing Unit, CPU) - выполнять команды;

Си. Команды процессора - примитивные операции, которые он может выполнять:

- команды выполняются одна за другой последовательно;
- каждая команда выполняет какую-то небольшую часть работы;
- команда выполняет операцию над операндами;

<sup>7а синтеза цифровых схем</sup>

• некоторые команды могут менять последовательность выполнения команд.

Школа синтеза Чифровых схем

• Последовательность команд, хранящаяся в памяти – **программа**.

чифровых схем

Школа синтеза цифров

з ча синтеза цифра

Школа синтеза чифро

Школа синтеза чифровачх схем

язык ассемьлера 14 ТЕ

<sup>7а синтеза Чифровых схем</sup>

### оли синтеза Цифров, АРХИТЕКТУРА НАБОРА КОМАНД

Школа синтеза Чифровых схем

Процессоры делятся на "семьи", в каждой из которых свой набор команд,
Каждый набор конкретного процессора реализует архитектуру набора команд (Instruction Set Architecture,

- ISA определяет команды с точностью до двоичной кодировки, поэтому процессоры из одной семьи могут выполнять одни и те же программы;
- Язык ассемблера (или просто ассемблер, англ. Assembly Language) язык программирования, прямо икола синтеза цифровых схем Дис Школа синтеза чифров <sup>7</sup>а синтеза Чифровых схем соответствующий ISA, но понятный человеку. 1HTe3a Чифровых схем

Школа синтеза цифров

з а синтеза цифра

Школа синтеза чифровых схем

ол и синтеза Цифровых схем

APXITEKTY PARISC-V

<sup>7а синтеза</sup> Чифровых схем

#### **RISC-V**

- Пятое поколение архитектур набора команд RISC, созданное в 2010 году исследователями из калифорнийского университета в Беркли;
- Спецификация ISA доступна для **свободного и бесплатного** использования Linux в мире архитектур;

з да синтеза Цифровых схем

- Предназначена для использования как в **коммерческих**, так и **академических** целях;
- Поддерживается общая растущая программная экосистема;
- Архитектура имеет **стандартную версию**, а также несколько **расширений** системы команд;
- Подходит для вычислительных систем всех уровней: от **микроконтроллеров** до **суперкомпьютеров**;
- Стандарт поддерживается некоммерческой организацией "RISC-V Foundation", которая работает в тесном партнерстве с "The Linux Foundation".

Школа синтеза Чифровых схем



APXIITEKT PARISC-V

## З за синтеза Цифровых схем RISC-V FOUNDATION



ола синтеза 4ифро

АССЕМБЛЕРИЈЕ В АССЕМБЛЕРЕ - РЕГИСТРЫ

# з да синтеза Чифровых схем

- В отличие от языков высокого уровня в ассемблере отсутствуют переменные
- Ола синтеза Вместо переменных команды оперируют с регистрами
  - место переменных комалды элдеги.

    ограниченный набор **ячеек хранения чисел**, встроенных прямо в аппаратуру формых схем
  - в архитектурах RISC арифметические операции могут выполняться только с регистрами
  - с памятью возможны только операции записи и считывания (в отличие от CISC)

Школа синтеза Чифровых схем

- л<sup>ОЛа</sup> синтеза цифровых схем Преимущество работы с регистрами - скорость доступа к ним <sup>1а синтеза</sup>,
  - Недостатки ограниченной число регистров 32 в RISC-V

<sup>7а синтеза Чифровых схем</sup>

Школа синтеза цифро

ола синтеза Цифро

Школа синтеза цифров

Школа синтеза цифров

Школа синтеза Чифров 81x схем

ACCEMENTER RISC-V

## ола синтеза Цифровых схем PEГИСТРЫ RISC-V

- 32 регистра для основного набора команд
- Oла синтеза (word)ровых сх х0 всегда равен О Каждый регистр имеет размер 32 бита = слово геза чифровых схем

<sup>7а синтеза Чифровых схем</sup>

- 32 регистра для вещественных операций в Ta c<sub>UHTe3a</sub>
  - 2 регистра дл...

    расширении "F"

    В версии RV64 регистры имеют размер 64 бита

_	Register	ABI Name	Description	Saver	]
основного набора команд	0	zero	· · · ·		
WK0/72 22 5	x1 WKO	ra	Return address Stack pointer	Caller	
имеет размер 32 бита = слово		spu <sub>HTe3a</sub>	Stack pointer	Callee	
умеет размер 32 онта – слово 4ифр <sub>ОВЫХ СХЕМ</sub>	x3	gp (C3a)	Global pointer	. 632	Moh
"TOPOBALLE	x4	tp	Thread pointer		40pp
on Cxen	x5	t0	Temporary/alternate link register	Caller	
<b>Y</b> /	x6-7	t1-2	Temporaries	Caller	
	x8	s0/fp	Saved register/frame pointer	Callee	
вещественных операций в	x9	s1	Saved register	Callee	
41/40	x10-11	a0-1	Function arguments/return values	Caller	
Ш <sub>КОЛА СИНТЕЗА</sub> егистры имеют размер 64 бита	x12-17	a2-7	Function arguments Saved registers	Caller	
THI COLD IN THE STATE OF THE ST	x18-27	s2-1123a	Saved registers	Callee	
егистры имеют размер 64 бита	x28-31	t3-6	Temporaries	Caller	ppos
егистры имеют размер 64 бита	f0-7	ft0-7	FP temporaries	Caller	NOB
XeM	f8-9	fs0-1	FP saved registers	Callee	
	f10-11	fa0-1	FP arguments/return values	Caller	
	f12-17	fa2-7	FP arguments	Caller	
	f18-27	fs2-11	FP saved registers	Callee	
4150-	f28-31	ft8-11	FP temporaries	Caller	
CNHT	, or la	CHH	FP temporaries $4/k_{OMA} c_{NH7}$ $10^{10}p_{OB}$ $9/X$ $6/k$		•
"11e3a ///		""/e <sub>3a</sub> //	• • • • • • • • • • • • • • • • • • • •	e3a ///	
Hudopop.		41	196 <sub>000</sub>	44	Poop
Школа синтеза Чифровых схем			FP temporaries  UKOMA CUHT  1\$\phi_{OBS}  YX \ CXEM		- 08
SYGW			NGW		

з ча синтеза цифра

зла синтеза Цифровых схе<sub>м</sub>

#### ACCEMENTER RISC-V ола синтеза Цифровых схем КОМАНДЫ RISC-V

Каждая команда имеет код операции (opcode) и операнды

жод операции (opcode) и операда # x1 = x2 + x3 # x1 = x3 + x3 # x1Школа синтеза цифро геза Чифровых схем

эла синтеза Цифровых схем

- add код операции сложение
- х1 регистр результата
- х2, х3 регистры-операнды
- используется в ассемблере для комментариев
- Ta CUHTe3a (source register):

  - Эквивалент в языке С:

<sup>7а синтеза Чифровых схем</sup>

Школа синтеза чифровію схем Школа синтеза цифров

з ча синтеза цифра

ACCEMBITER RISC-V SI'a CNHTe3a 4Nchpobi зла синтеза цифровых схем з за синтеза цифра АРИФМЕТИЧЕСКИЕ КОМАНДЫ

- rd, rs1, rs2 add
- геложение rd = rs1 + rs2  $c_{UHTe3a}$   $4u_{dposblx}$   $c_{XeM}$ 

  - - побитовое И rd = rs1 & rs2
- rd, rs1, rs2 Hresa Huppobbly Cxem ог та, чифопобитовое ИЛИ rd = rs1 | rs2

  - rd, rs1, rs2 xor
    - побитовое исключающее ИЛИ rd = rs1 xor rs2
  - rd, rs1, rs2
- <sup>7а синтеза Чифровых схем</sup>

Школа синтеза Чифровых схем Школа синтеза цифро

Школа синтеза *Чифровых* схем Школа синтеза 4ифров

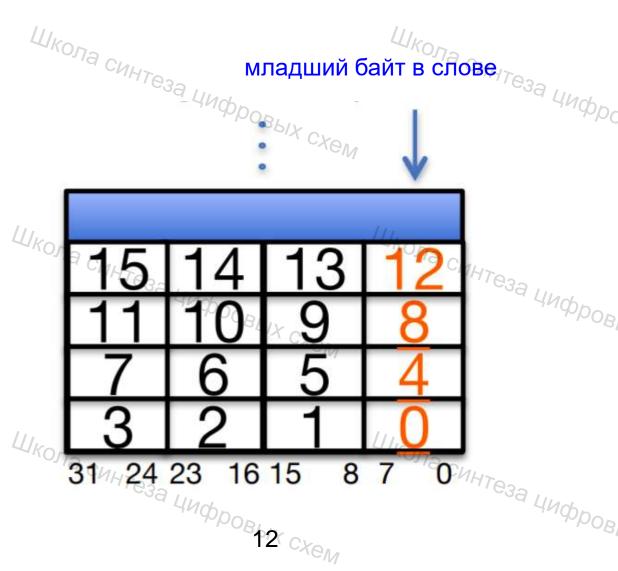
Школа синтеза чифрових схем Школа синтеза цифров ACCEMENTER RISC-V ол синтеза Цифровых схем

### ОБРАЩЕНИЕ К ПАМЯТИ

- 1 байт = 8 бит
- 4 байта года адреса в памяти адреса в байтах
- Школа синтеза чифровых схел в RISC-V байты в словах расположены в соответствии с

atle endian, т.е. ос. в младших битах (см. картинку) интеза чифровых схем <sup>7</sup>а синтеза Чифровых схем

Школа синтеза Чифровых схем <sup>7а синтеза Чифровых схем</sup>



з ча синтеза цифра

од и синтеза Цифровых схем

## З З Синтеза Цифровых схем КОМАНДЫ ОБРАЩЕНИЯ К ПАМЯТИ

- rd, addr
  - Чкола синтеза Чифровых схем считывание слова в регистр rd из памяти по адресу addr
- rd, addr запись слова в из регистра rd в память по адресу addr
- также доступны обращения меньшими размерами:
  - halfword 2 байта: Ih, sh
- Школа синтеза Чифровых схем □ byte - 1 баит: ID, SD интеза ■ адрес addr может быть указан несколькими способами, самый простой

где **r1** - регистр, содержащий адрес обращения

указагно-w x2, 4(x3) # считывание слова из адреса = x3 + 40 ла синтеза чифрова 43 схем

<sup>7а синтеза цифровых схем</sup> лнтеза чифровых схем Школа синтеза цифро

з на синтеза цифр<sub>с</sub>

Школа синтеза цифров

Школа синтеза цифров

## з да синтеза Цифровых схем ЧИСЛОВЫЕ КОНСТАНТЫ

- В коде ассемблера в качестве операндов можно использовать числа или непосредственные значения Школа синтеза цифро
  - (immediate operand) или константы
- (пппеснате орегано) изиказивания:

  Синтез Для использования констант в архитектуре предусмотрены специальные команды: Чифровых схем
  - addi<sup>OB</sup><sub>b/x</sub> rd, rs1, imm
    - сложение rd = rs1 + imm, например
    - x2, x3, -4 addi
- # x2 = x3 4
- 7a <sub>Cинтеза</sub> rd, imma CM

<sup>7а синтеза Чифровых схем</sup>

- ы запись константы в регистр rd = rs1 чист
- числа по умолчанию указываются в десятичной системе и со знаком
- Школа синтеза *Чифровых* схем для использований шестнадцатиричных чисел нужно добавить "0x", например, 0x10 = 16

Школа синтеза Чифровых схем

Школа синтеза цифров

Школа синтеза цифров

Sia CNHTe3a 4Ndpo

зла синтеза Цифровых схем

Школа синтеза Чифров 14 схем

#### SI'a CNHTe3a UNCHPOBL З З Синтеза Цифровых схем ВЕТВЛЕНИЯ В ПРОГРАММЕ. ПЕРЕХОДЫ

- Ветвления подразумевают, что в зависимости от результатов некоторых вычислений нужно выполнять Школа синтеза цифро
  - разные действия
- В языках программирования используется оператор if
- Аналог оператора if в ассемблере операции условного перехода (branch)

Школа синтеза Чифровых схем

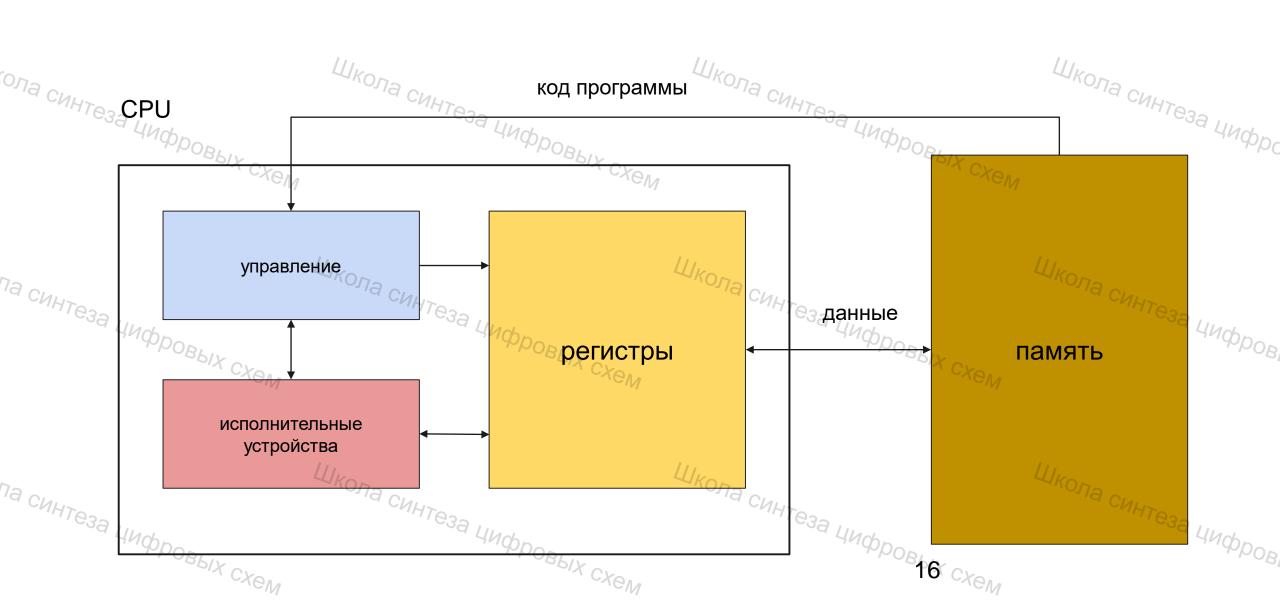
- rs1, rs2, label # branch if equal beq
- если rs1 == rs2, сделать переход на участок кода, помеченный label, иначе выполнить следующую Школа синтеза *Чифров* ne rs1, rs2, label # branch if not equal eсли rs1!= rs2, сделать переход на участок кода, помеченный label, иначе выполнить следующую Ta cultresta
  - bne
  - команду
  - также есть безусловные переходы јитр
- label <sup>7а синтеза цифровых схем</sup>

зла синтеза цифр<sub>с</sub>

Школа синтеза цифров

заключениеровь **УИТЕ** З ча синтеза цифровь З ч синтеза чифровь зла синтеза цифр<sub>с</sub>

### АРХИТЕКТУРА С ТОЧКИ ЗРЕНИЯ ПРОГРАММИСТА



#### ола синтеза Цифровь SI'a CUHTE3a LINCOPPOBL ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ АССЕМБЛЕРА

Зачем программировать на ассемблере, если есть языки высокого уровня?

Школа синтеза Чифровых схем

программировать на ассемблере, если есть языки высокого уровни.
Ассемблер до сих пор используется в системном ПО (например, ОС), чтобы получить чкола синтеза чифро Ола синтеза Чифг Синтеза чифровых схем

зла синтеза 4ифро

Школа синтеза цифров

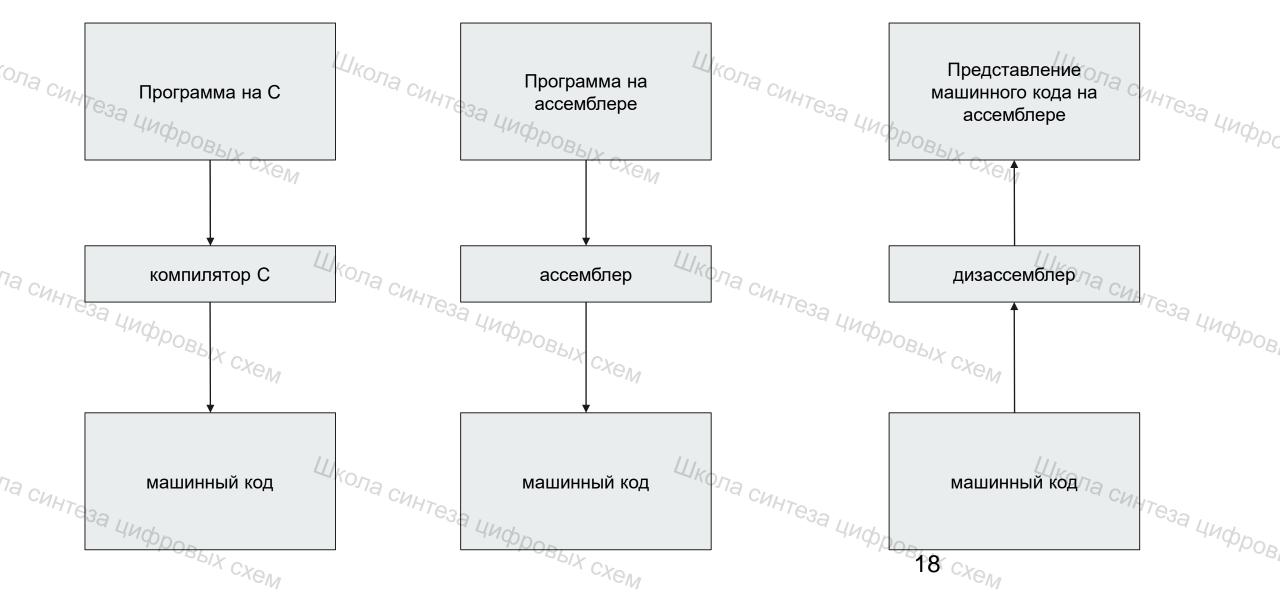
- Ассемблер используется при разработке аппаратуры:
- для написания тестовых программ

<sup>7а синтеза Чифровых схем</sup>

для изучения особенностей работы аппаратуры при выполнении программ используют Школа синтеза цифров дизассемблирование, т.е. получение из двоичного кода ассемблерной программы знис, эла синтеза Чифровых схем Alac Jila Cuhtesa Luppobbix cxem <sup>7</sup>а синтеза Чифровых схем

Школа синтеза Чифроват схем

# заключение ровь ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ АССЕМБЛЕРА СХЕМ



ола синтеза 4ифро

дополнительная информация ола синтеза Цифровых схем ПСЕВДОИНСТРУКЦИИ

з а синтеза цифровых схем Команды ассемблера, которые упрощают читаемость, но при сборке в двоичный код заменяются на # no operation

# ster

\*\*ster Школа синтеза цифро  $\Leftrightarrow addi x0, x0, 0$ CUHTE3a NOP другие  $\Leftrightarrow$  addi rd, rs,  $\mathbf{0}^{86/\chi}$   $c_{\chi_{\Theta_M}}$ mv rd, rs begz rs, offset  $\Leftrightarrow$  beg rs, x0, offset # branch if = zero Школа синтеза Чифровых схем Школа синтеза *Чифровых* схем Школа синтеза Чифров <sup>7а</sup> синтеза Чифровых схем

> Школа синтеза Чифров 19 схем Школа синтеза Чифровых схем

Школа синтеза 4ифров

зла синтеза цифр<sub>с</sub>

<sup>7а синтеза</sup> Чифровых схем

дополнительная информация

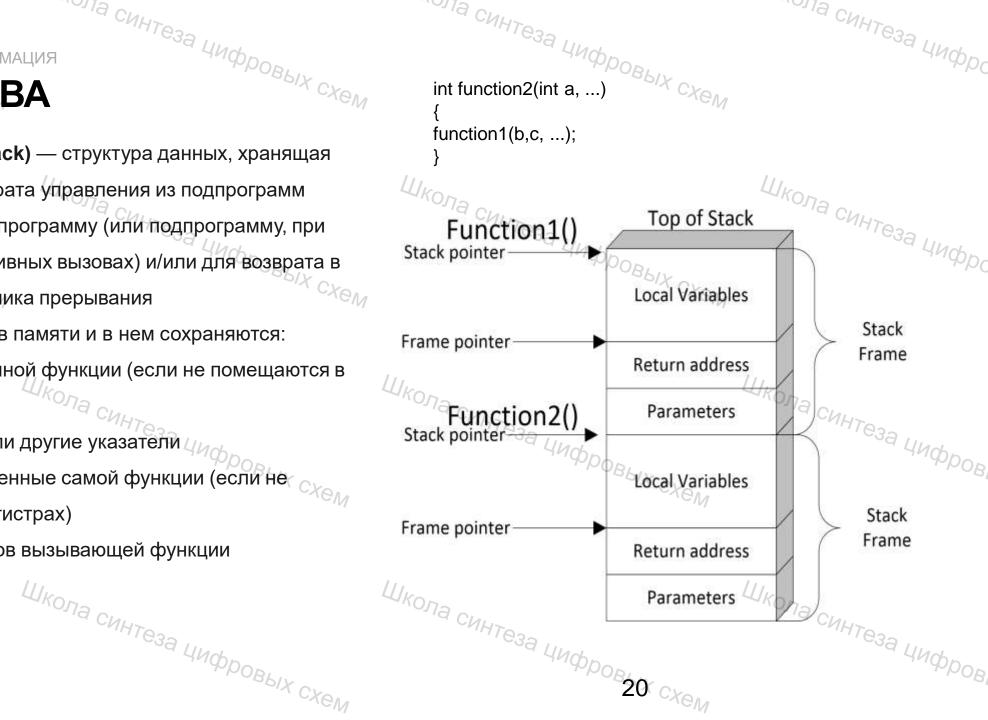
#### СТЭК ВЫЗОВА

<sup>7а синтеза цифровых схем</sup>

Стек вызовов (call stack) — структура данных, хранящая информацию для возврата управления из подпрограмм (процедур, функций) в программу (или подпрограмму, при вложенных или рекурсивных вызовах) и/или для возврата в программу из обработчика прерывания

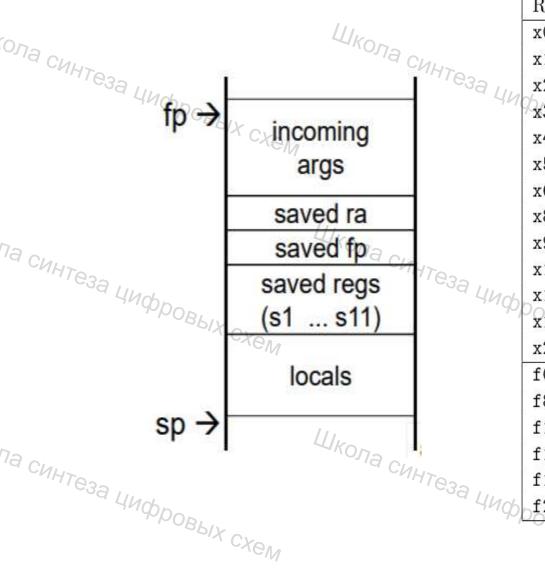
Стек обычно хранится в памяти и в нем сохраняются:

- аргументы вызванной функции (если не помещаются в регистрах)
- адрес возврата или другие указатели
- локальные переменные самой функции (если не помещаются в регистрах)
- значения регистров вызывающей функции



дополнительная информация

## дополнительная информация РЕГИСТРЫ СТЭКА ВЫЗОВА СХЕМ



6			-	· ·	1
	Register	ABI Name	Description	Saver	
1	x0	zero //	Hard-wired zero	$-U_{K}$	0.5
	x1	ra	Return address	Caller	OJIA CHHT
LIIA	x2	sp	Return address Stack pointer Global pointer Thread pointer	Callee	ола <sub>синтеза</sub> чифро
70/(	x2 x30 <sub>Bb/X</sub> x4	gp	Global pointer	_	77/900
	x4	Ctp	Thread pointer	, —	
	x5	t0	Temporary/alternate link register	Caller	
	x6-7	t1-2	Temporaries	Caller	
	x8	s0/fp	Saved register/frame pointer	Callee	
Mapp	x9	so/ip s1 UKO	Saved register	Callee	70
	x10-11	a0-1	Function arguments/return values	Caller	CNHTOS
	x12-17	a2-7	Function arguments	Caller	440h
	x18-27 s2-11		Saved registers Temporaries	Callee	<sup>Та синтеза</sup> 4ифров
	x28-31	t3-6	Temporaries	Caller	
	f0-7	ft0-7	FP temporaries	Caller	
	f8-9	fs0-1	FP saved registers	Callee	
	f10-11	fa0-1 ///	FP arguments/return values	Caller	
	f12-17	fa0-1 fa2-7	FP arguments	Caller	72 0
	f18-27	fs2-11	FP saved registers	Callee	CNHTE32
496	f28-31	ft8-11	FP temporaries	Caller	Та си <sub>НТеза</sub> Цифров
. /⊑	086/X CX	· · · · · · · · · · · · · · · · · · ·	FP saved registers FP temporaries 1000000000000000000000000000000000000	,	. POB
	9/	GW	XeM		

ола синтеза Цифровых схем

зла синтеза цифр<sub>с</sub>

дополнительная информация

Пакаја

## ол синтеза Цифровых схем КОМАНДЫ JAL, JALR, RET

- Команды JAL/JALR выполняют безусловный переход с сохранением текущего РС для возможности возврата в Школа синтеза цифро тот же участок программы

- jal rd, label # jump and link синтеза чифо сделать переход на участок кода, помеченный label (immediate address), и записать (рс+4) в регистр rd
  - rd, offset(rs1) # jump and link register jalr
- сделать переход на участок кода по адресу (rs1 + offset), и записать (pc+4) в регистр rd jalr x0, x14,0 e3a 4upposbix cxem <sup>1а</sup> синтеза 4ифровых схем

<sup>7а синтеза Чифровых схем</sup>

# return from subroutine: эза чифровых схем

Школа синтеза Чифровых схем

зла синтеза Чифровых схем

зла синтеза цифр<sub>с</sub>

Школа синтеза 4ифров

Школа синтеза цифров

 $U_{KOЛA}$  синTe3a Чифров22 схем

дополнительная информация ола синтеза Цифровых схем од на синтеза Цифровых схем олы синтеза цифр<sub>с</sub>

https://github.com/riscv/riscv-isa-manual/releases/download/Ratified-IMAFDQC/riscv-spec-20191213.pdf - RISC-V Школа синтеза цифро

ISA Specification (unprivileged)

https://godbolt.org/ - Compiler Explorer

https://syntacore.com/page/products/sw-tools - Pre-built tools

https://syntacore.com/page/products/sw-tools - Pre-built tools

https://syntacore.com/page/products/sw-tools - Pre-built tools Чифровых схем

Школа синтеза Чифровых схем Школа синтеза *Чифровых* схем Школа синтеза 4ифров

Школа синтеза цифровых схем <sup>7а</sup> синтеза Чифровых схем

> $U_{KOЛ}$ а синтеза Чифров $_{23}$  схем Школа синтеза 4ифров

Школа синтеза Чифровых схем <sup>7а синтеза Чифровых схем</sup>