

# Процессорное ядро schoolMIPS

Young Russian Chip Architects  
summer school

# Введение

- **Микроархитектура:**  
аппаратная реализация архитектуры в виде схемы
- **Процессор:**
  - **Тракт данных:** функциональные блоки обработки и передачи данных (арифметико-логическое устройство, регистровый файл, мультиплексоры и т.д.)
  - **Устройство управления:** формирует управляющие сигналы для функциональных блоков

Application Software	programs
Operating Systems	device drivers
Architecture	instructions registers
Micro-architecture	datapaths controllers
Logic	adders memories
Digital Circuits	AND gates NOT gates
Analog Circuits	amplifiers filters
Devices	transistors diodes
Physics	electrons

# Микроархитектура

- Возможны несколько аппаратных реализаций одной и той же архитектуры:
  - **Однотактная реализация:** каждая инструкция выполняется за один такт
  - **Многотактная реализация:** каждая инструкция разбивается на несколько шагов и выполняется за несколько тактов
  - **Конвейерная реализация:** каждая инструкция разбивается на несколько шагов и несколько инструкций выполняются одновременно

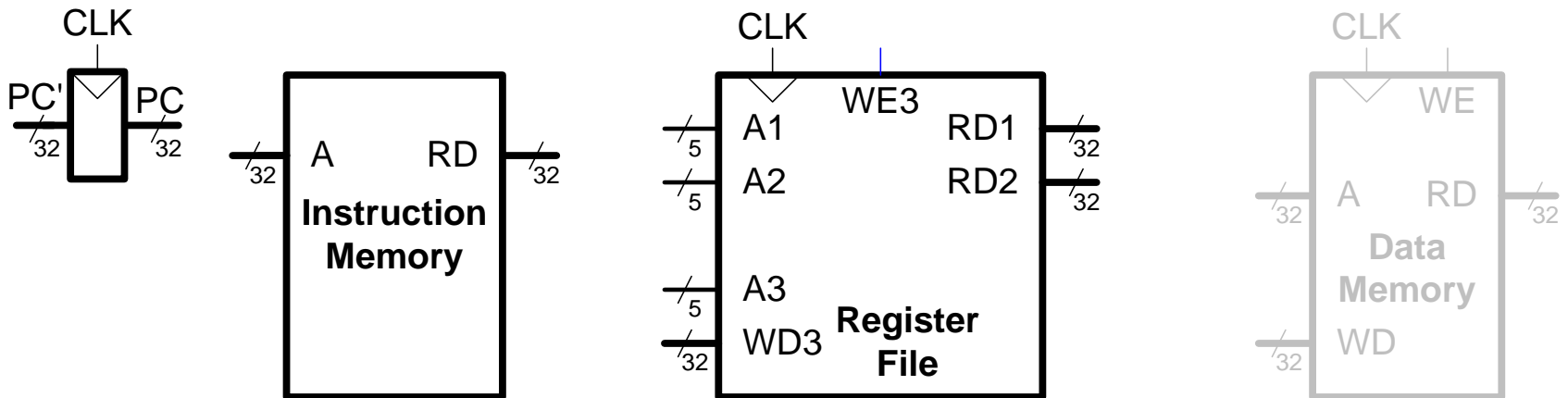
# MIPS процессор schoolMIPS

- Однотактная реализация
- Отсутствует память данных
- Словная адресация памяти инструкций
- Инструкции:
  - R-типа (оба аргумента хранятся в регистрах):  
addu, or, srl, sltu, subu
  - I-типа (один из аргументов - константа):  
addiu, lui
  - I-типа (инструкции ветвления):  
beq, bne

# Архитектурное состояние

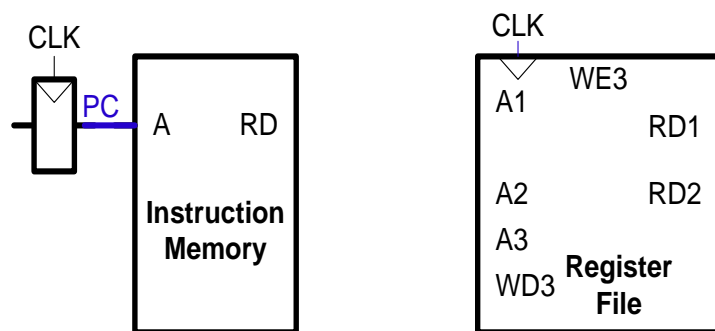
- Определяется:
  - Содержимым счетчика команд (РС)
  - Содержимым 32-х регистров общего назначения
  - Содержимым памяти (команд, данных)

# Элементы, хранящие состояние MIPS



# Процессор schoolMIPS: инструкция addiu

**Шаг 1:** Выборка (считывание) инструкции addiu из памяти

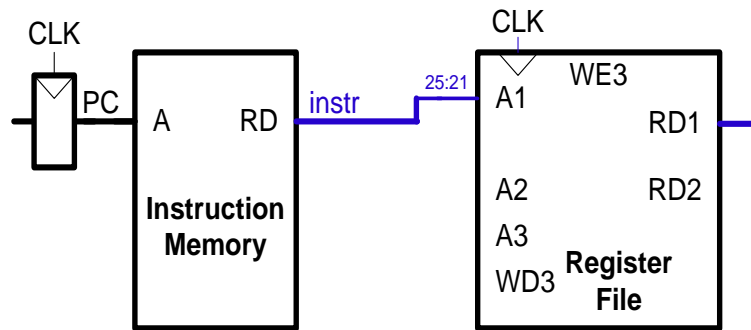


**I-type.** Integer Add Immediate,  $rt = rs + \text{Immediate}$

31	op	26	25	rs	21	20	rt	16	15	Immediate	0
----	----	----	----	----	----	----	----	----	----	-----------	---

# Процессор schoolMIPS: инструкция addiu

**Шаг 2:** считывание операндов-источников из регистрового файла



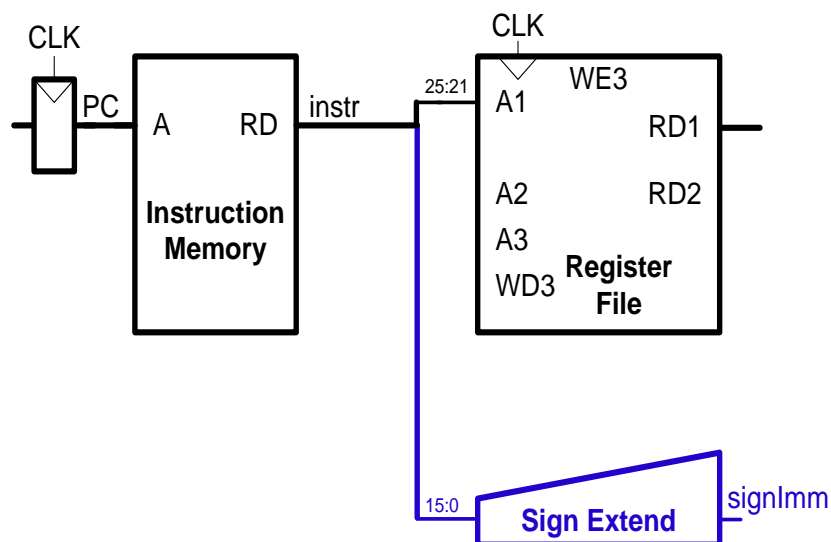
**I-type.** Integer Add Immediate,  $rt = rs + \text{Immediate}$

31	op	26	25	rs	21	20	rt	16	15	Immediate	0
----	----	----	----	----	----	----	----	----	----	-----------	---



# Процессор schoolMIPS: инструкция addiu

**Шаг 3:** расширение 16-битной константы до 32-х разрядов битом знака

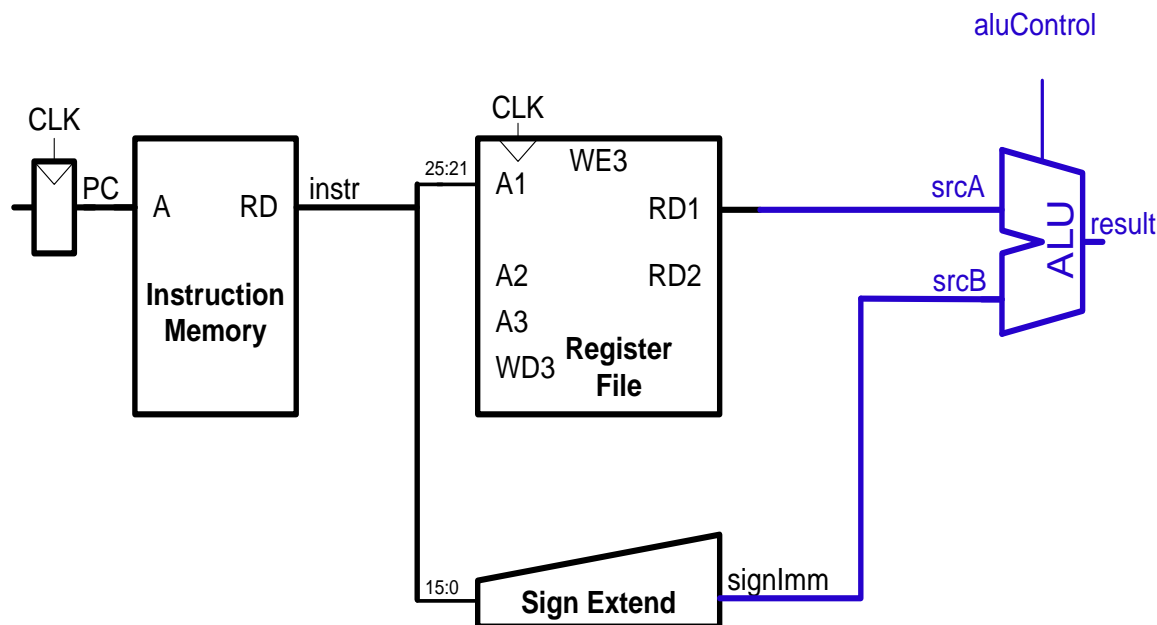


**I-type.** Integer Add Immediate,  $rt = rs + \text{Immediate}$

31	op	26	25	rs	21	20	rt	16	15	Immediate	0
----	----	----	----	----	----	----	----	----	----	-----------	---

# Процессор schoolMIPS: инструкция addiu

**Шаг 4:** вычисление результата арифметической операции

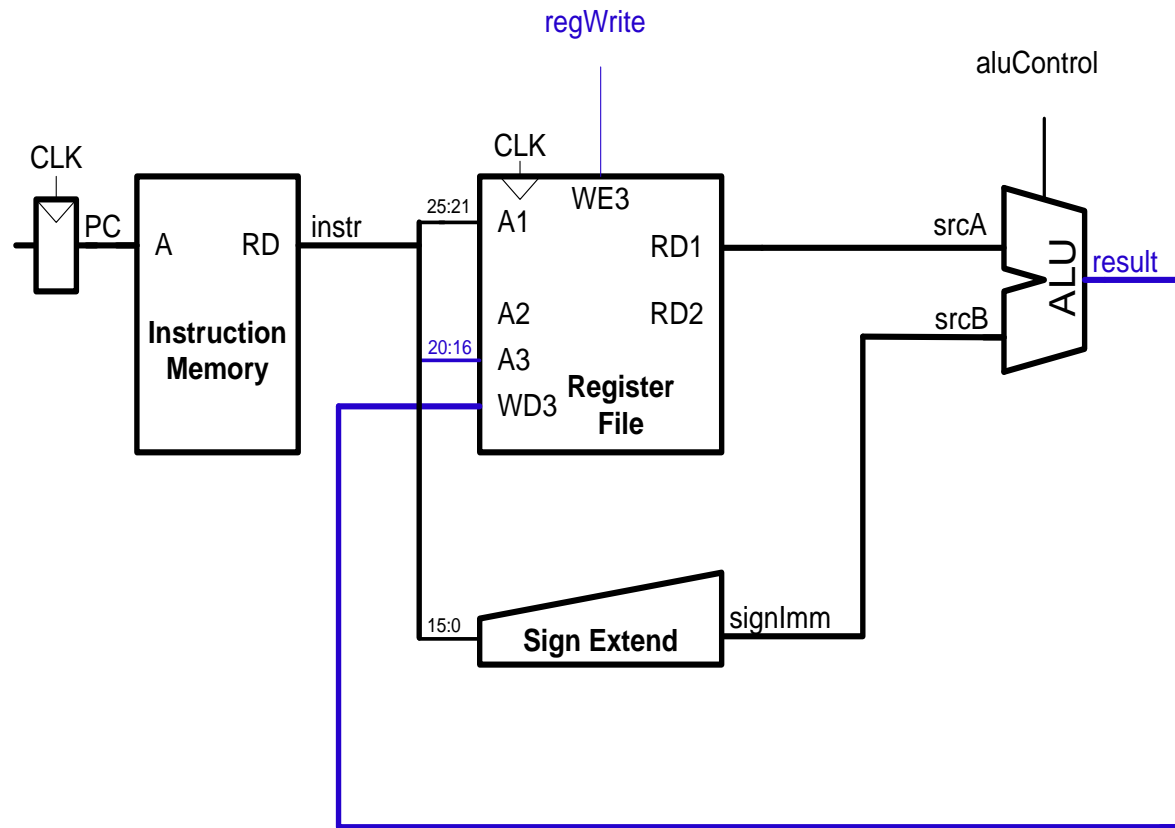


**I-type.** Integer Add Immediate,  $rt = rs + \text{Immediate}$

31	op	26	25	rs	21	20	rt	16	15	Immediate					0
----	----	----	----	----	----	----	----	----	----	-----------	--	--	--	--	---

# Процессор schoolMIPS: инструкция addiu

## Шаг 5: запись результата вычислений в регистр

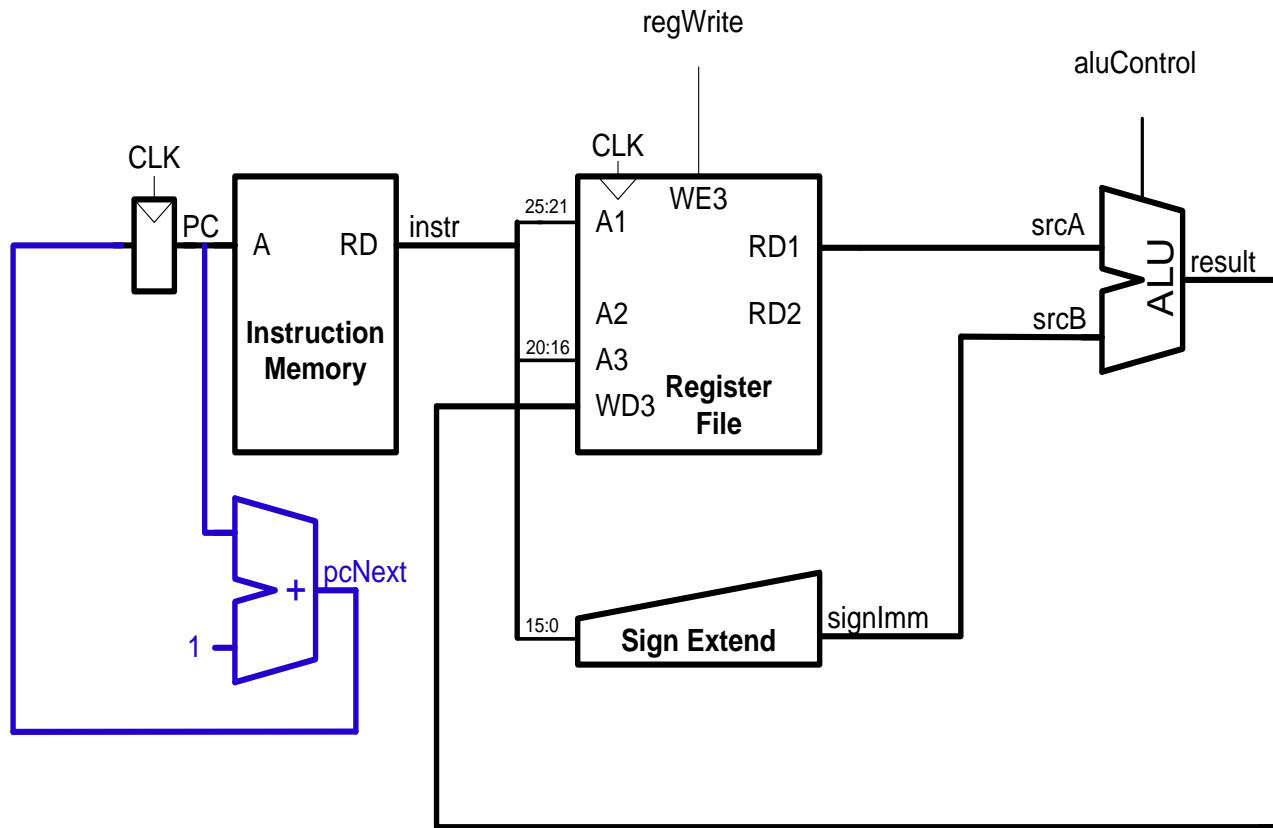


**I-type.** Integer Add Immediate,  $rt = rs + \text{Immediate}$

31 <b>op</b> 26	25 <b>rs</b> 21	20 <b>rt</b> 16	15	<b>Immediate</b>	0
-----------------	-----------------	-----------------	----	------------------	---

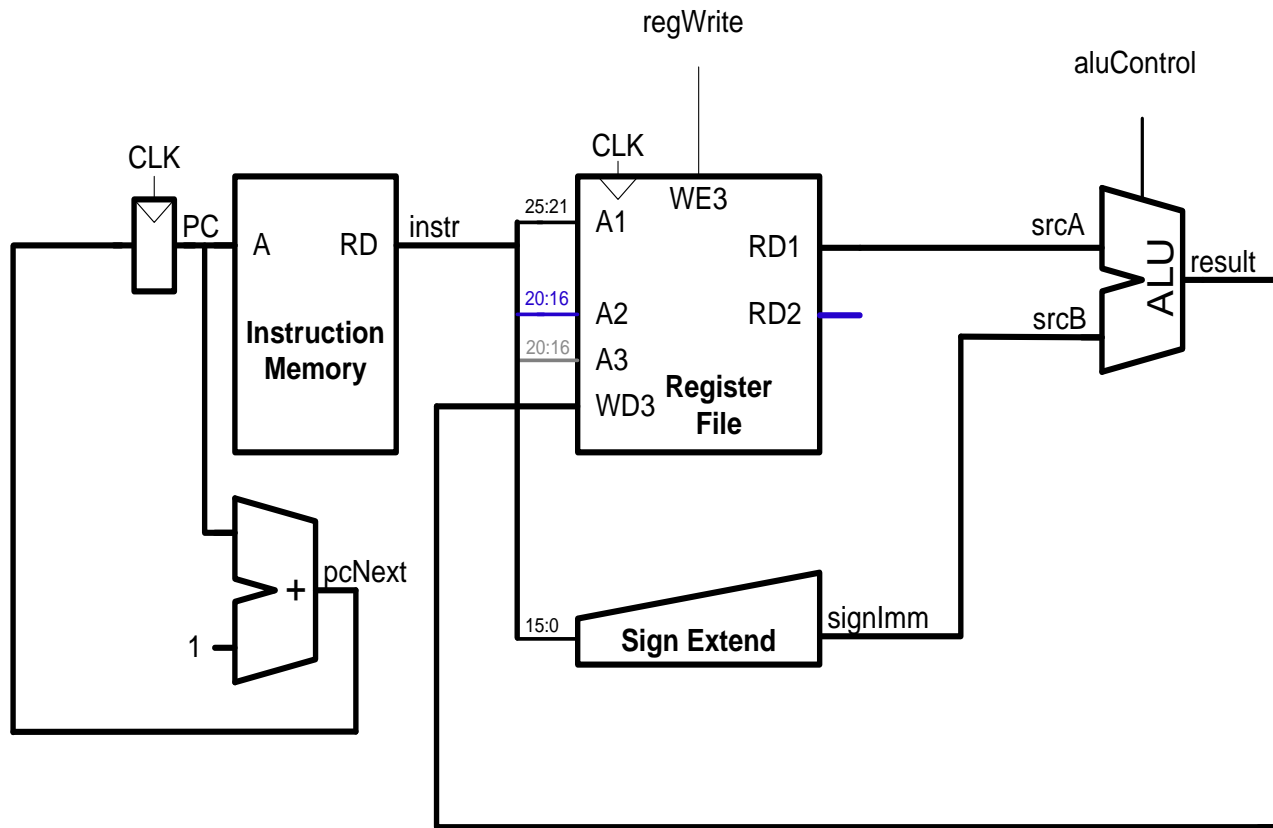
# Процессор schoolMIPS: инструкция addiu

**Шаг 6:** вычисление адреса следующей инструкции



# Процессор schoolMIPS: инструкция addu

- считывание операнда 2 из регистрового файла

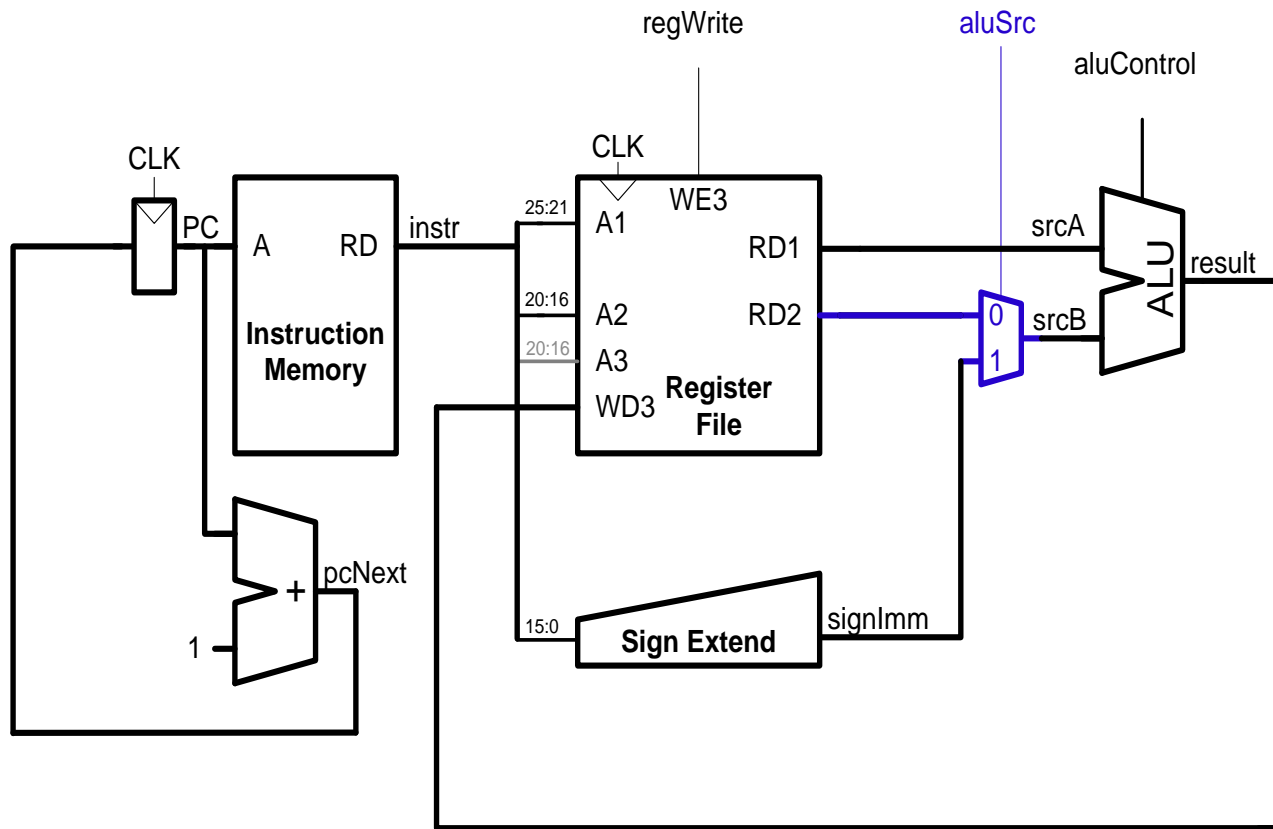


**R-type.** Integer Add Unsigned,  $rd = rs + rt$

31	op	26	25	rs	21	20	rt	16	15	rd	11	10	sa	6	5	funct	0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	-------	---

# Процессор schoolMIPS: инструкция addu

- передача данных операнда 2 в арифметико-логическое устройство

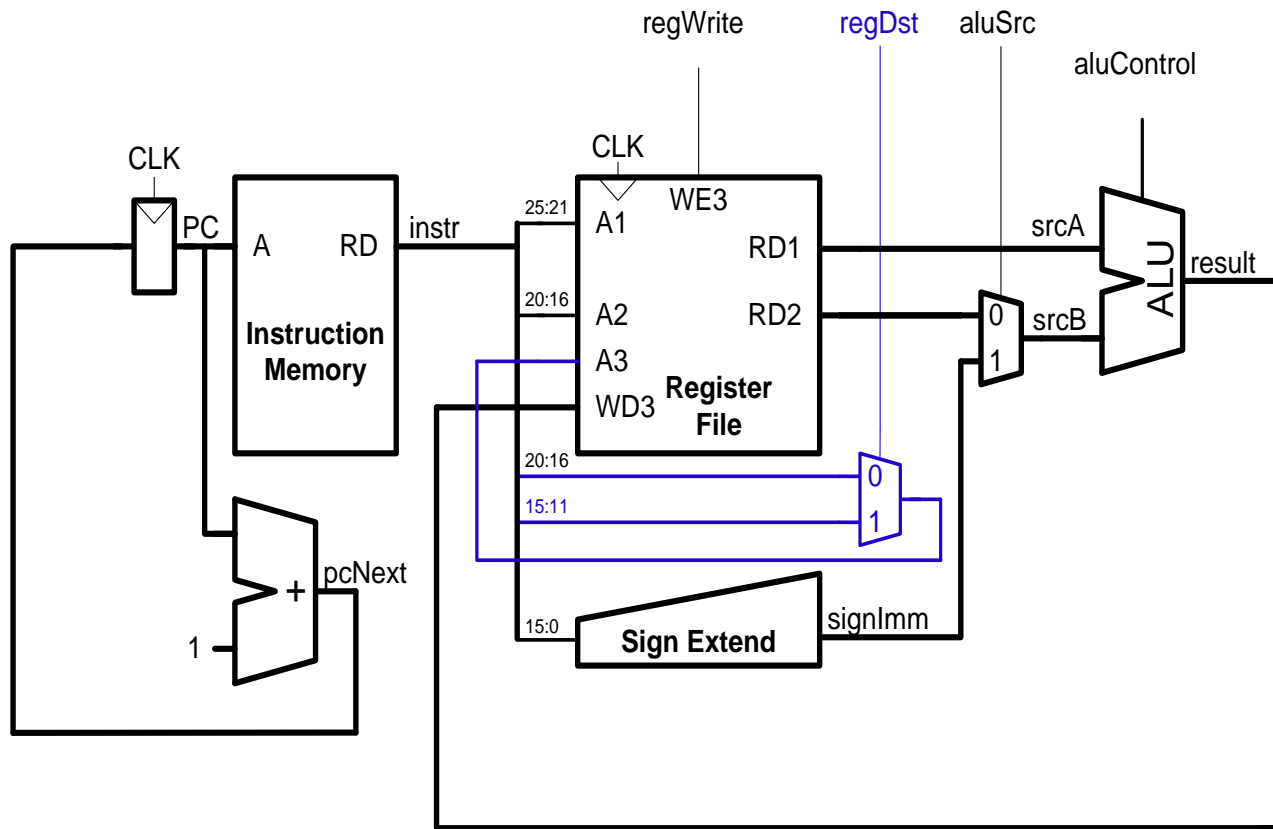


**R-type.** Integer Add Unsigned,  $rd = rs + rt$

31	op	26	25	rs	21	20	rt	16	15	rd	11	10	sa	6	5	funct	0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	-------	---

# Процессор schoolMIPS: инструкция addu

- определение регистра для записи результата
- запись результата вычислений

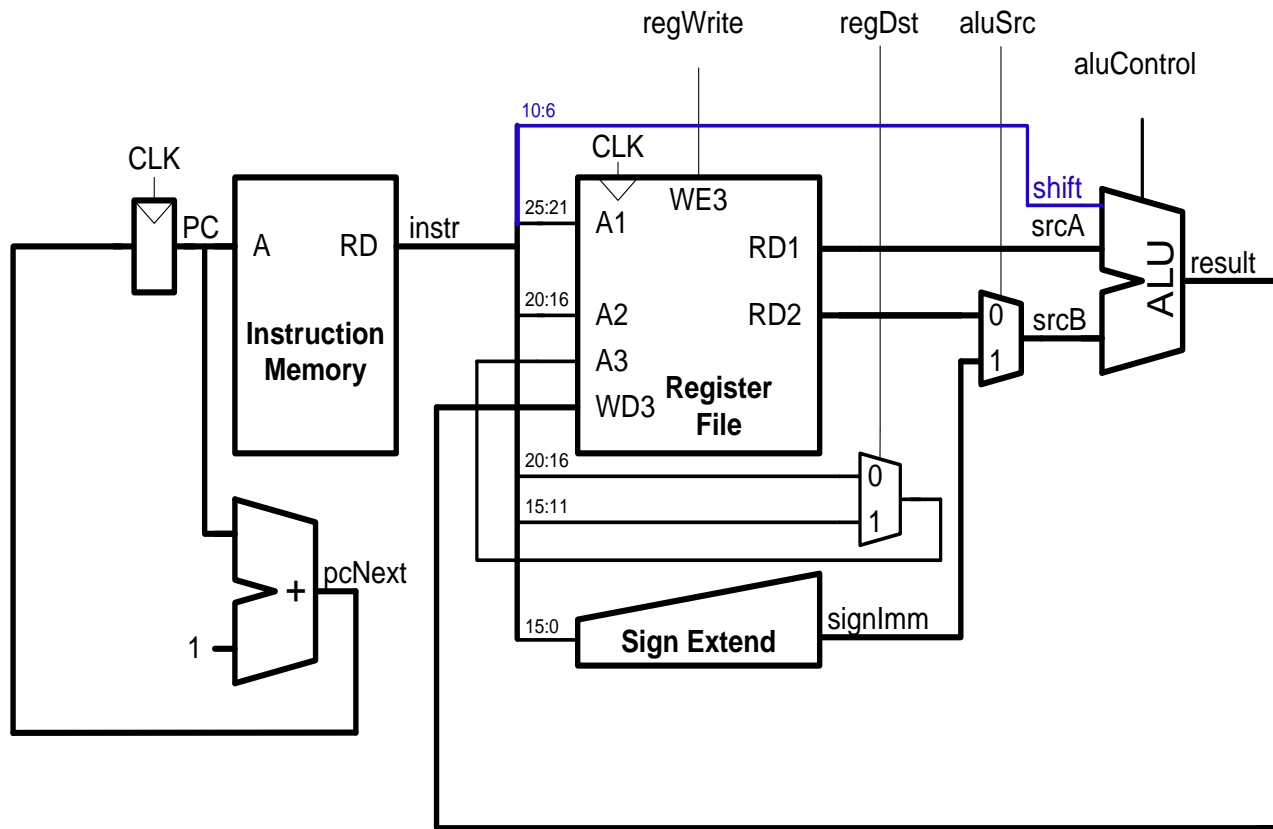


**R-type.** Integer Add Unsigned,  $rd = rs + rt$

31	op	26	25	rs	21	20	rt	16	15	rd	11	10	sa	6	5	funct	0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	-------	---

# Процессор schoolMIPS: инструкция srl

- передача данных о размере сдвига в арифметико-логическое устройство



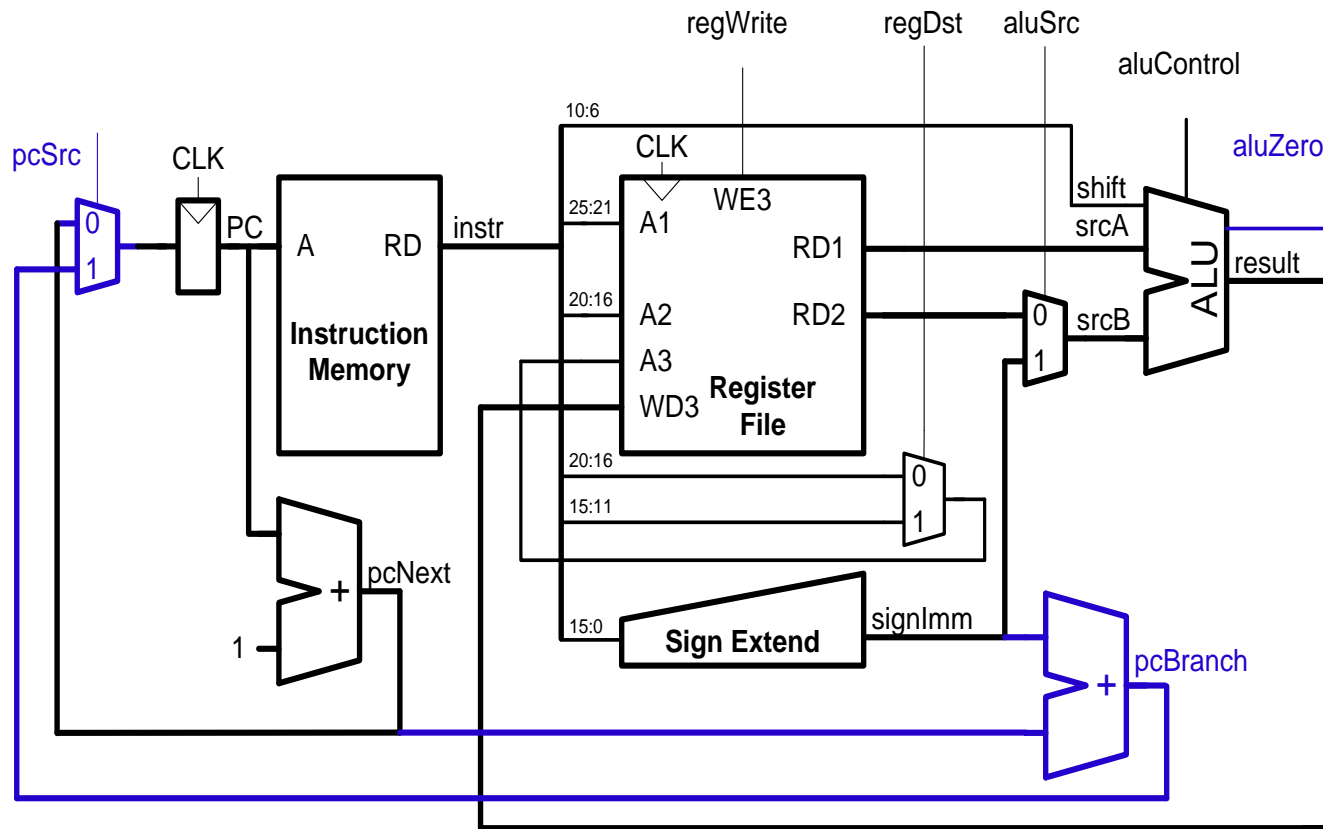
**R-type.** Shift Right Logical,  $rd = (uns)rt \gg sa$

31	op	26	25	rs	21	20	rt	16	15	rd	11	10	sa	6	5	funct	0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	-------	---



# Процессор schoolMIPS: инструкция beq

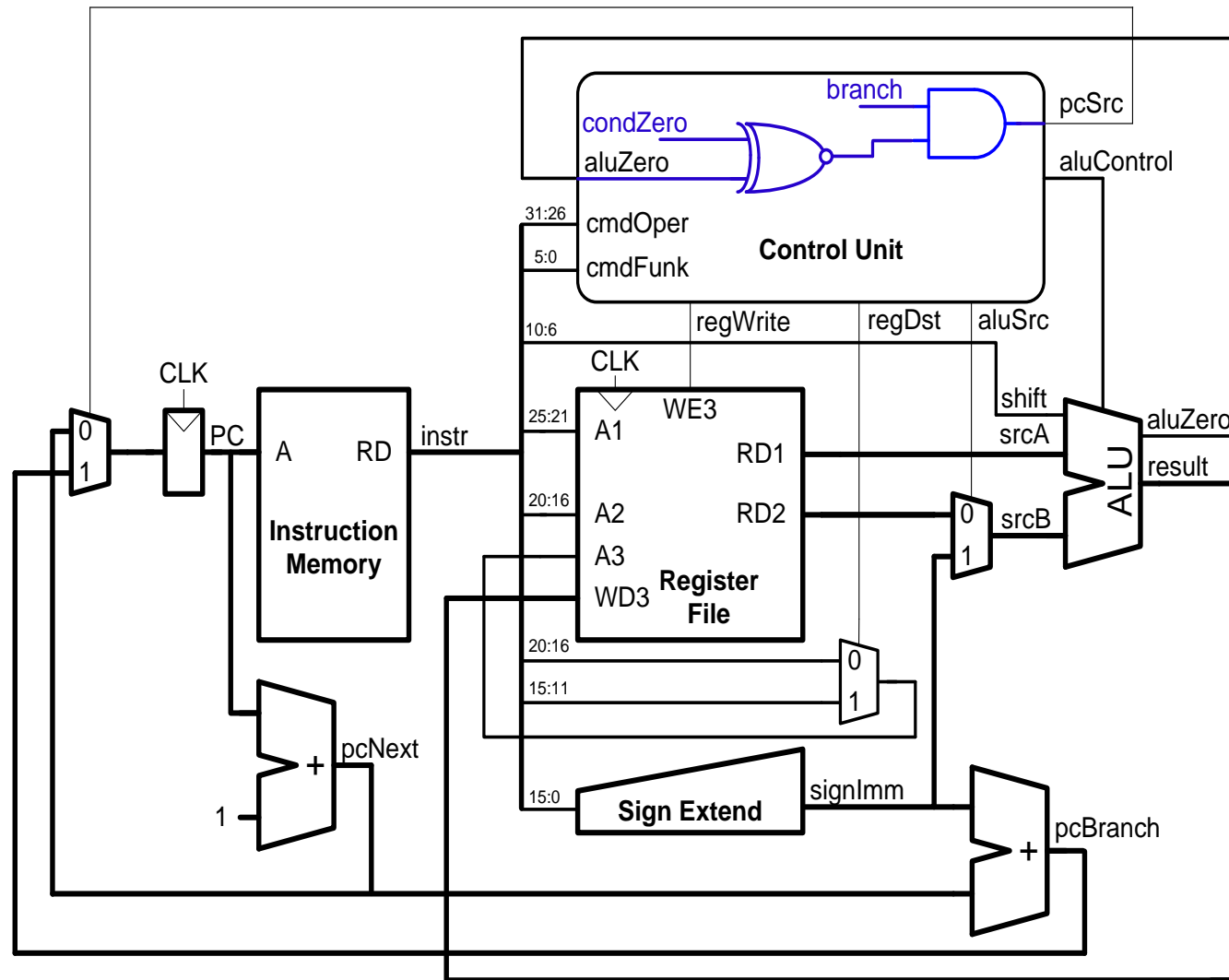
- вычисление адреса следующей инструкции



**I-type.** Branch On Equal, if ( $R_s == R_t$ )  $PC += (int)offset$

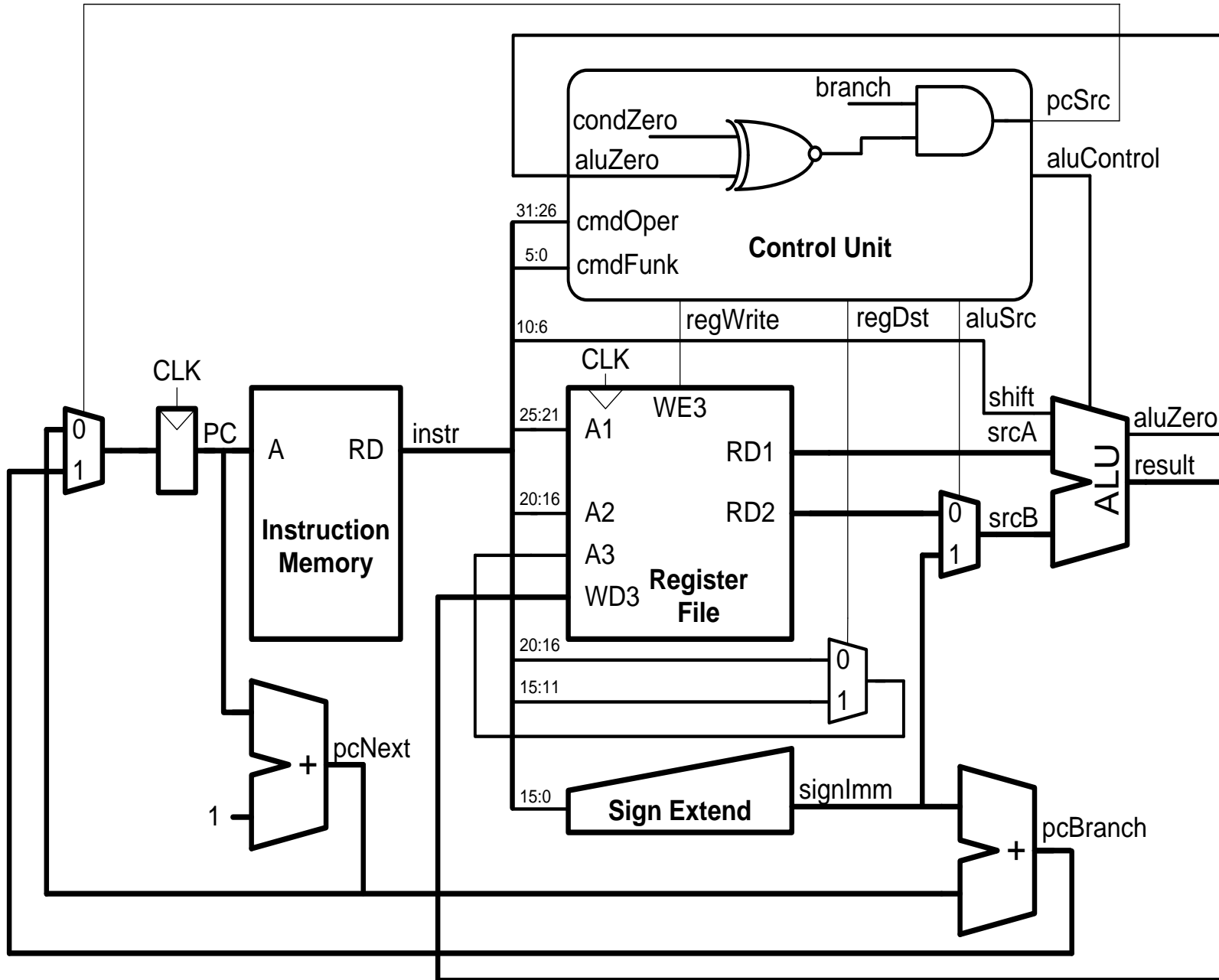
31	op	26	25	rs	21	20	rt	16	15	Immediate					0
----	----	----	----	----	----	----	----	----	----	-----------	--	--	--	--	---

# Процессор schoolMIPS: инструкция beq



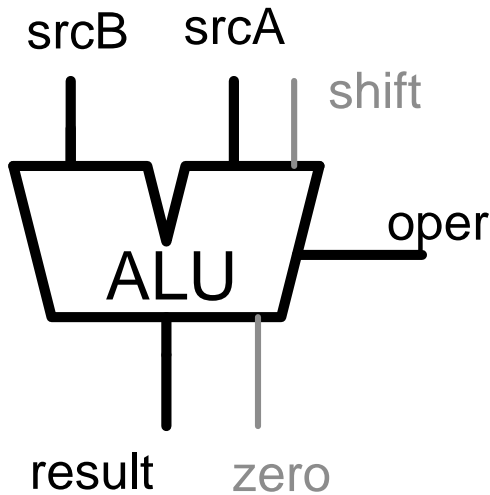
- определение необходимости перехода в зависимости от равенства результата нулю

# Процессор schoolMIPS: итоговая схема



# Процессор schoolMIPS:

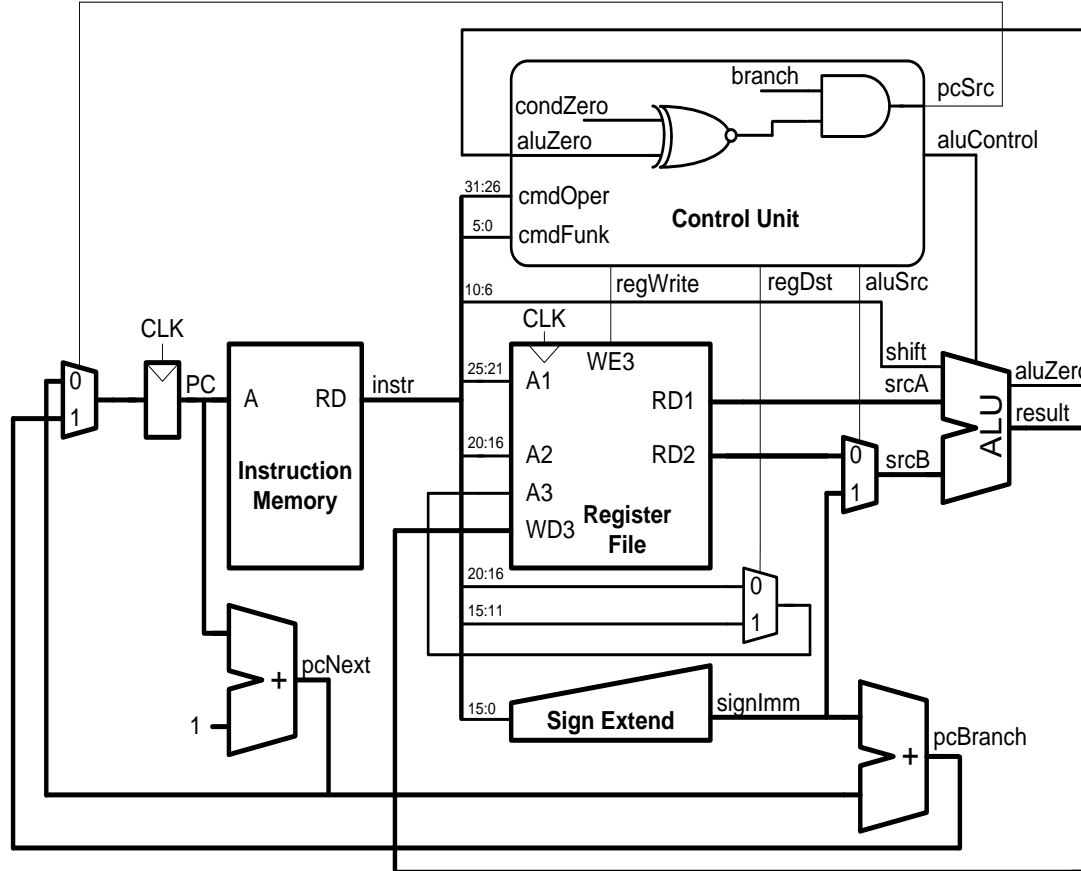
## Арифметико-логическое устройство



oper <sub>2:0</sub>	Функция	Описание
000	ADD	$A + B$
001	OR	$A \mid B$
010	LUI	$B \ll 16$
011	SRL	$B \gg \text{shift}$
100	SLTU	$(A < B) ? 1 : 0$
101	SUBU	$A - B$
110	Не исп.	
111	Не исп.	

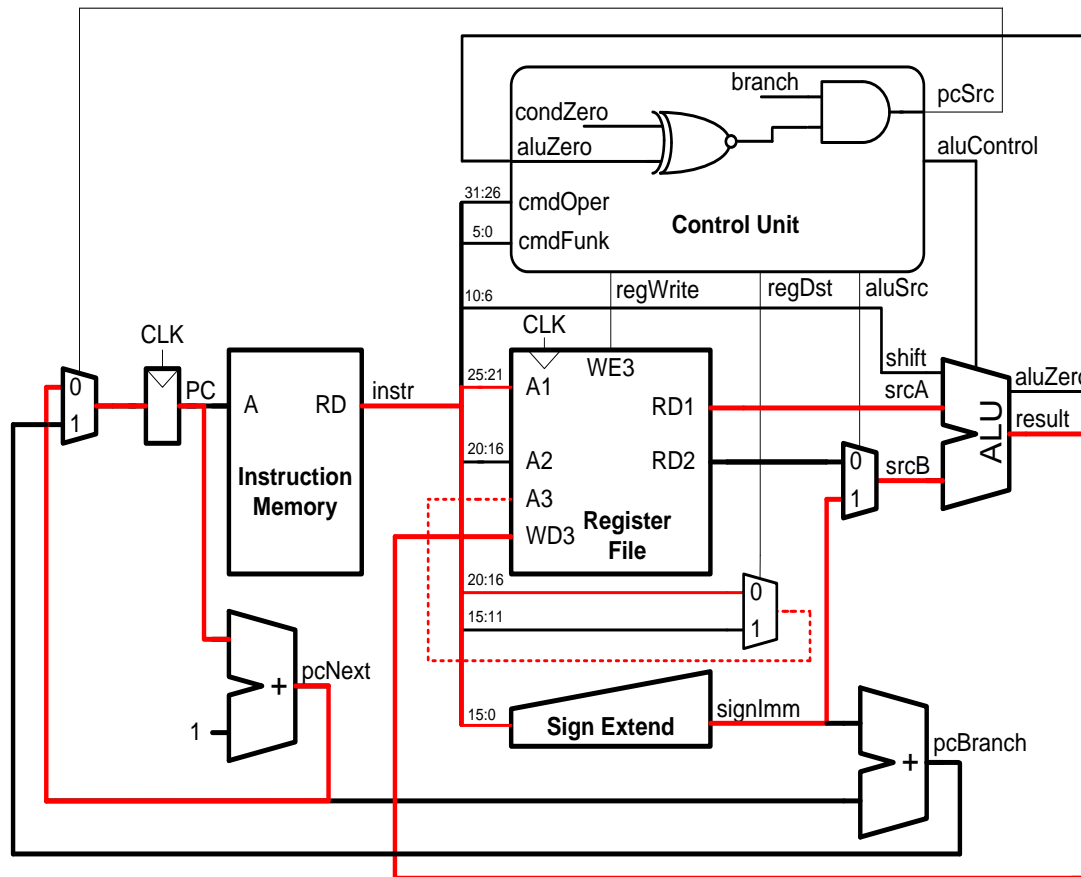
# Процессор schoolMIPS: сигналы управления

Instr	cmdOper	cmdFunc	branch	condZero	regDst	regWrite	aluSrc	aluControl



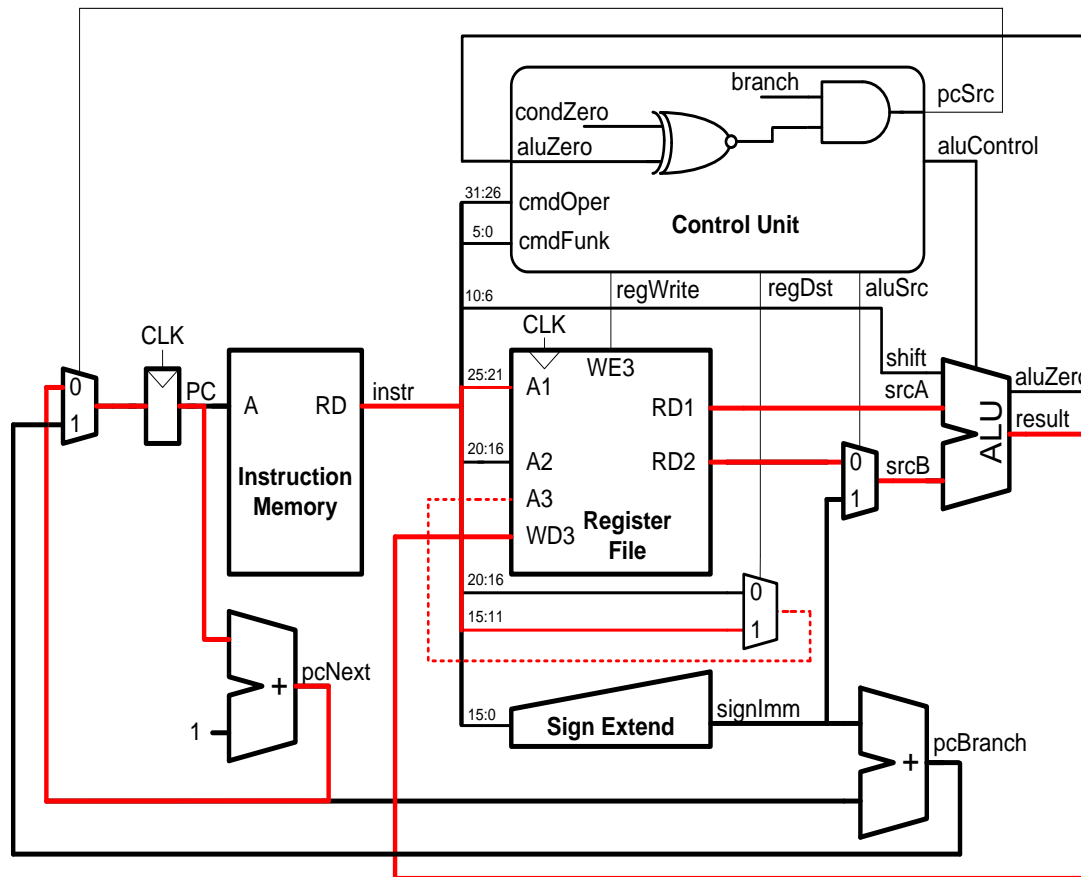
# Процессор schoolMIPS: сигналы управления

Instr	cmdOper	cmdFunk	branch	condZero	regDst	regWrite	aluSrc	aluControl
addiu	001001	??????	0	0	0	1	1	000



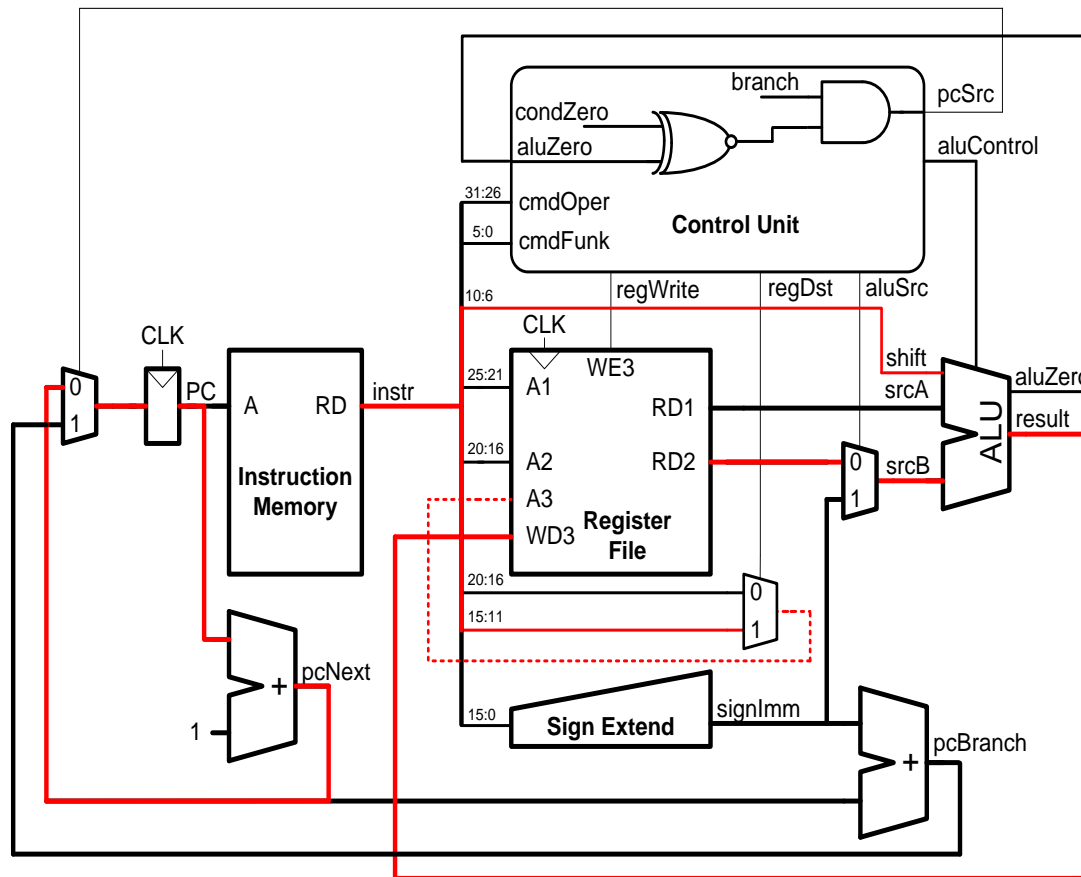
# Процессор schoolMIPS: сигналы управления

Instr	cmdOper	cmdFunc	branch	condZero	regDst	regWrite	aluSrc	aluControl
addiu	001001	??????	0	0	0	1	1	000
<b>addu</b>	<b>000000</b>	<b>100001</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>000</b>



# Процессор schoolMIPS: сигналы управления

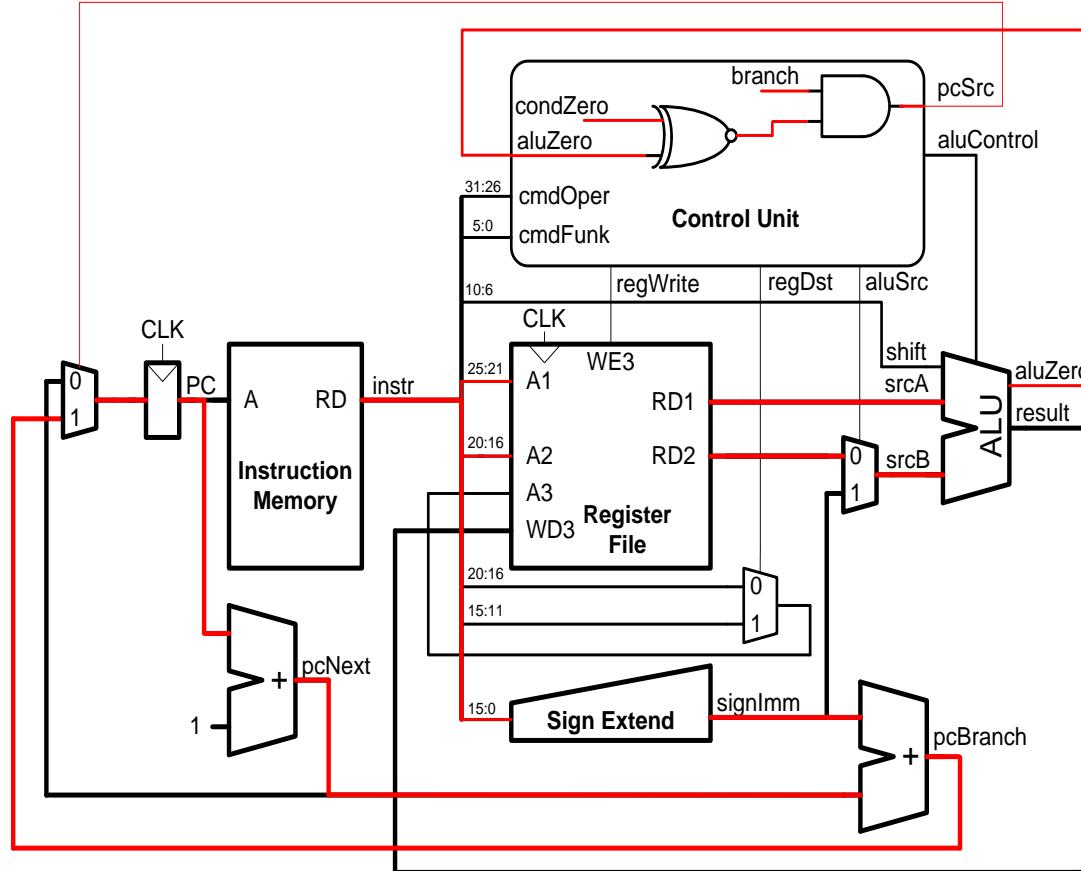
Instr	cmdOper	cmdFunc	branch	condZero	regDst	regWrite	aluSrc	aluControl
addiu	001001	??????	0	0	0	1	1	000
addu	000000	100001	0	0	1	1	0	000
srl	000000	000010	0	0	1	1	0	011





# Процессор schoolMIPS: сигналы управления

Instr	cmdOper	cmdFunc	branch	condZero	regDst	regWrite	aluSrc	aluControl
addiu	001001	??????	0	0	0	1	1	000
addu	000000	100001	0	0	1	1	0	000
srl	000000	000010	0	0	1	1	0	011
beq	000100	??????	1	1	0	0	0	000



Ваши вопросы?