Лаб. Упражнение No: 1

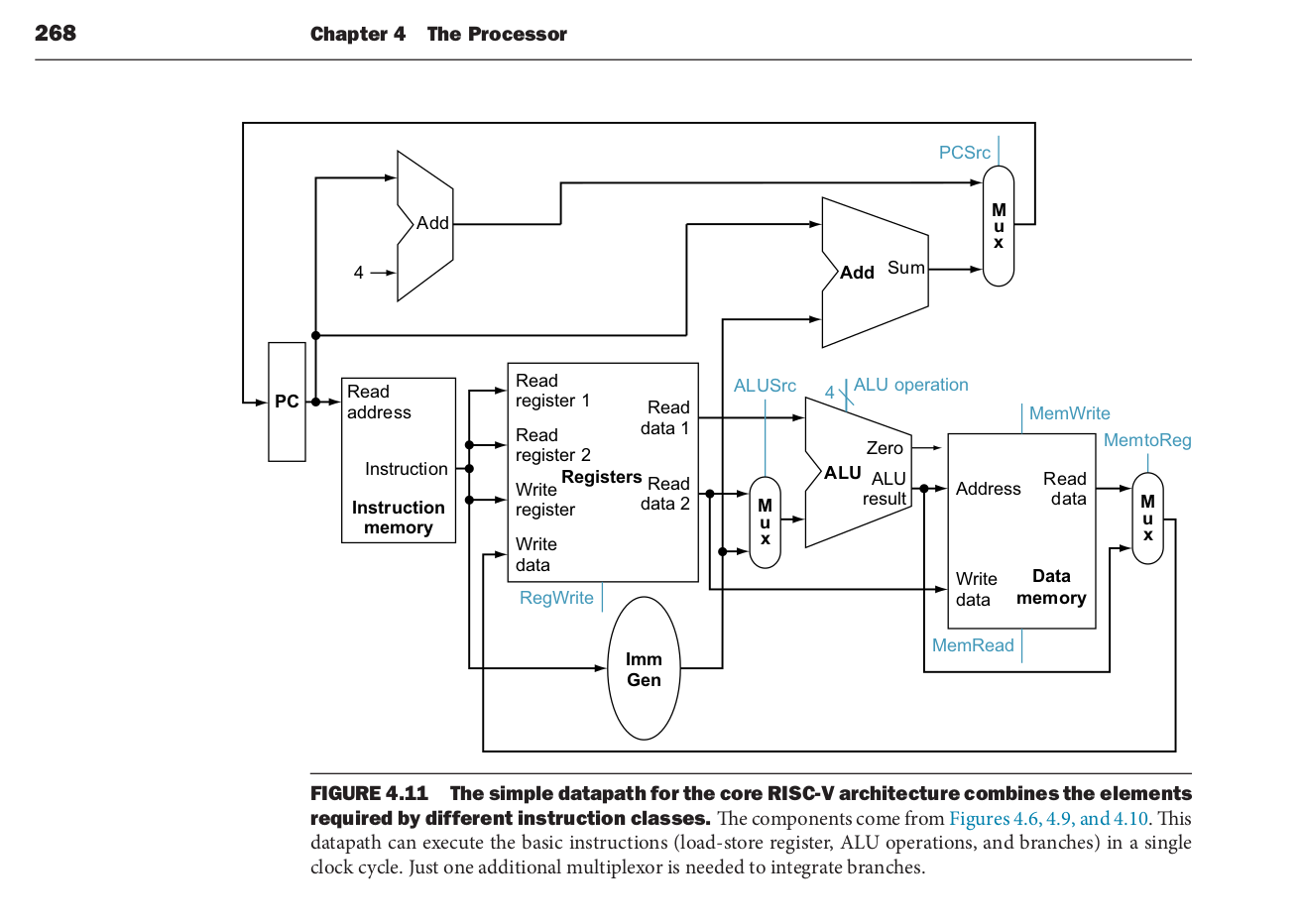
Дисциплина: Компютърни Архитектури

Асистент: Иван Янчев

========================================================================

1. **Теория:**

* ISA( Instruction Set Architecture) – Стандарт за инструкционно множество, което процесорите могат да изпълняват. Процесори с един ѝ същ ISA могат да изпълняват един ѝ същ код.
* RISC( Reduced Instruction Set Computing) – MIPS, RISC-V, ARM   
  Еднаква дължина на инструкциите в битове.  
  Малък брой изпълними инструкции.  
  По-малко средно време за изпълнение на инструкция.  
  Кодът заема повече памет.
* CISC( Coplex Instruction Set Computing) – x86, x86-64  
  Различна дължина на инструкциите в битове.  
  Голям брой изпълними инструкции.  
  Някои инструкции могат да бъдат разбити на няколко RISC инструкции.  
  Кодът е по-сбит.
* Datapath – Циклична схема на елементите и техните връзки, нужни за изпълнението на инструкциите от ISA.



Фиг 1. - Пример за RISC-V datapath, взет от страница 268 от учебника.

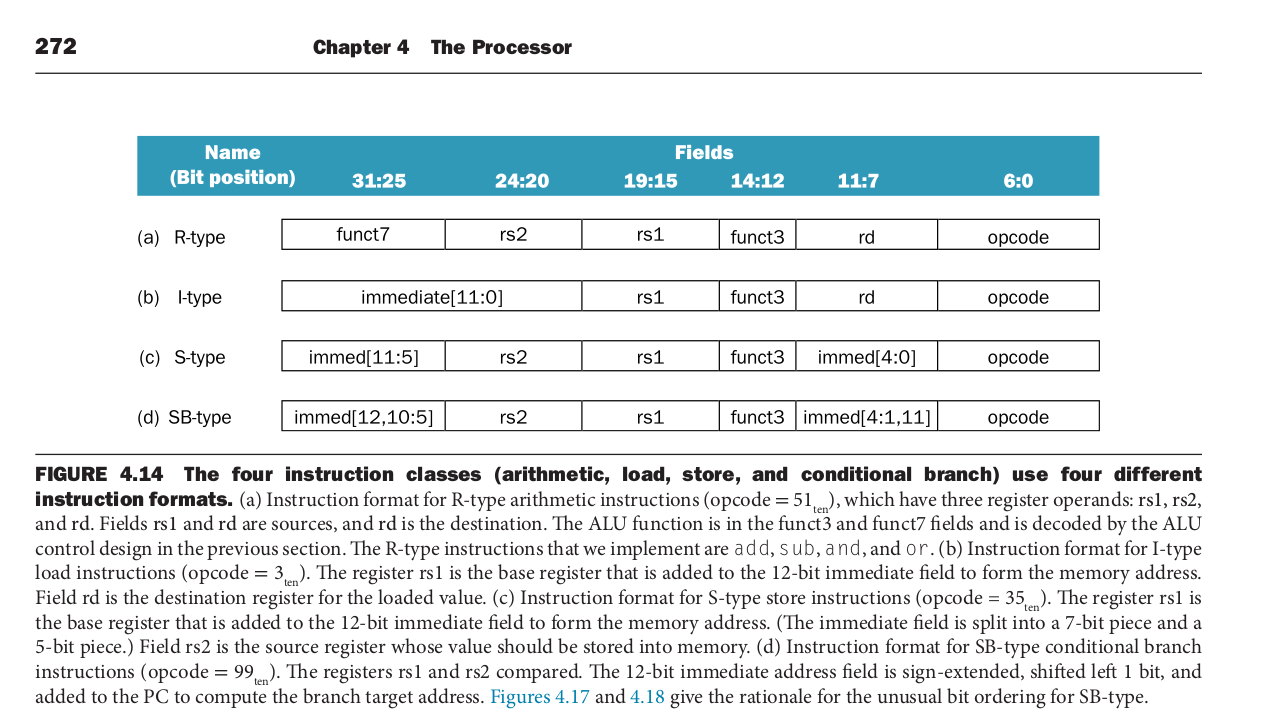
1. **Елементи:**Program Counter( PC) - Регистър / “брояч”, пазещ адресът на изпълняващата се в момента инструкция. Служи за следене на до къде е стигнала програмата в изпълнението си.

Instruction Memory – Памет в която се помества компилираната програна( инструкции в машинен код), които да бъдат изпълнени. На Фиг 1. се работи с 32-битови инструкции.   
  
Registers – Регистров файл намиращ се на самият процесор и е най-бързата памет. От регистрите се взимат данните за изпълнение на инструкциите.  
  
ALU( Arithmetic Logic Unit) – Аритметично-логическо устройство извършващо различни аритметично-логически операции. Пример: Събиране, Изваждане, Логическо ИЛИ и т.н.  
  
Data Memory – Памет в която се поместват данните на програмата. Пример: променливи, константи, т.н.  
  
Add( Adder) – Осакатена и опростена версия на ALU, която може да извършва само операцията събиране.  
  
Imm Gen( Immediate generator) – Разширява някаква бинарна стойност до битовия размер с който работи архитектурата( 32 на Фиг 1). Добавя 0-ли пред положителните числа( вкл. 0) и 1-ци пред отрицателните докато не се достигне желаният размер.

Mux( Multiplexor) – Устройство, което от няколко входни сигнала избира един и го препраща на изхода. Входните сигнали могат да бъдат многобитови. “ От няколко стойности избира една”.

1. **Типове инструкции за часа:**  
   R-Type – Аритметично-логически инструкции, използващи два регистъра за четене и един за запис.  
     
   I-Type – Подобни на R-Type, но вместо два регистъра за четене се използва един, а втората стойност за исканата операция се взима директно от битовете на инструкцията. Този формат се използва и за load( lw) инструкцията.  
     
   S-Type – Инструкция използваща два регистъра за четене и една immediate стойност. Нейната семантика е записване на стойността намираща се в първия регистър в адрес от паметта( data memory), намиращ се във втория регистър. Към стойността на втория регистър се добавя отместването( immediate).   
   Пример:  
   В регистър rs1 е запазена стойността 15, която искаме да запишем на адрес 2002 в паметта( data memory). Този адрес се намира в регистър rs2. Отместването ( immediate) е 4, тогава на адрес 2006( rs2 + immediate) от data memory ще бъде записано 15(rs1).

SB-Type – Инструкция за условен преход. използва два регистъра за четене и една immediate стойност. Стойностите запазени в двата регистъра се сравняват и спрямо резултатът от сравнението се преминава, или не се преминава на новия branch адрес. Използва се при if, for, while и т.н. конструкции.   
Пример: Ако условието е вярно в if-а, се изпълнява тялото на if-a (x на брой инструкции), ако не е, тези x на брой инструкции се прескачат, отива се на друг адрес.  
Новият адрес се изчислява като към стойността намираща се в pc се добави immediate стойността.   
Пример: В PC сме на адрес 10000 и immediate-а е 100. Ако условието за преход е успешно, новият адрес е 10000 + 100. SB инструкцията, която ще се разгледа този час е beq( Branch on equal), при която преходът се взима, ако стойностите са еднакви.



Фиг 2. Схема на четири RISC-V инструкционни формата и тяхното битово разпределение, което няма да бъде разглеждано в този час.

1. **Допълнителна теория:**Контролни сигнали – сините линии, служат за избор на път.  
   Специални регистри - регистри със специална цел.
2. **За протокола:**

Начертайте datapath-а за инструкциите Add( R-Type), Store( S-Type), Load( I-type) и beq( SB-Type), обяснете принципа им на работа, както и разликте между тях. **Не чертайте елементи, които не се използват от конкретната инструкция.** **Тези елементи се пропускат.**

1. **Линкове**

**Упражнения**: [https://github.com/tu-iyan/Computer-Architecture-2024](упражнения)

**Учебник:** [http://library.lol/main/373C67B0C5E22C9B92B1D8FACDC47E68](учебник)

**Чат ГПТ:** [https://chat.openai.com/](AI)

**Гугъл:** [https://www.google.com/](google)