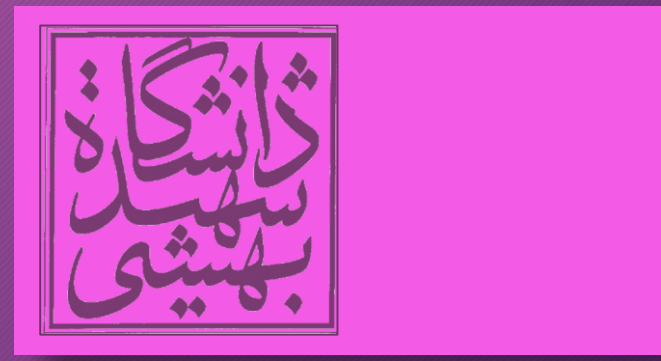
پردازنده MIPS Pipe Line (Hazard Free)

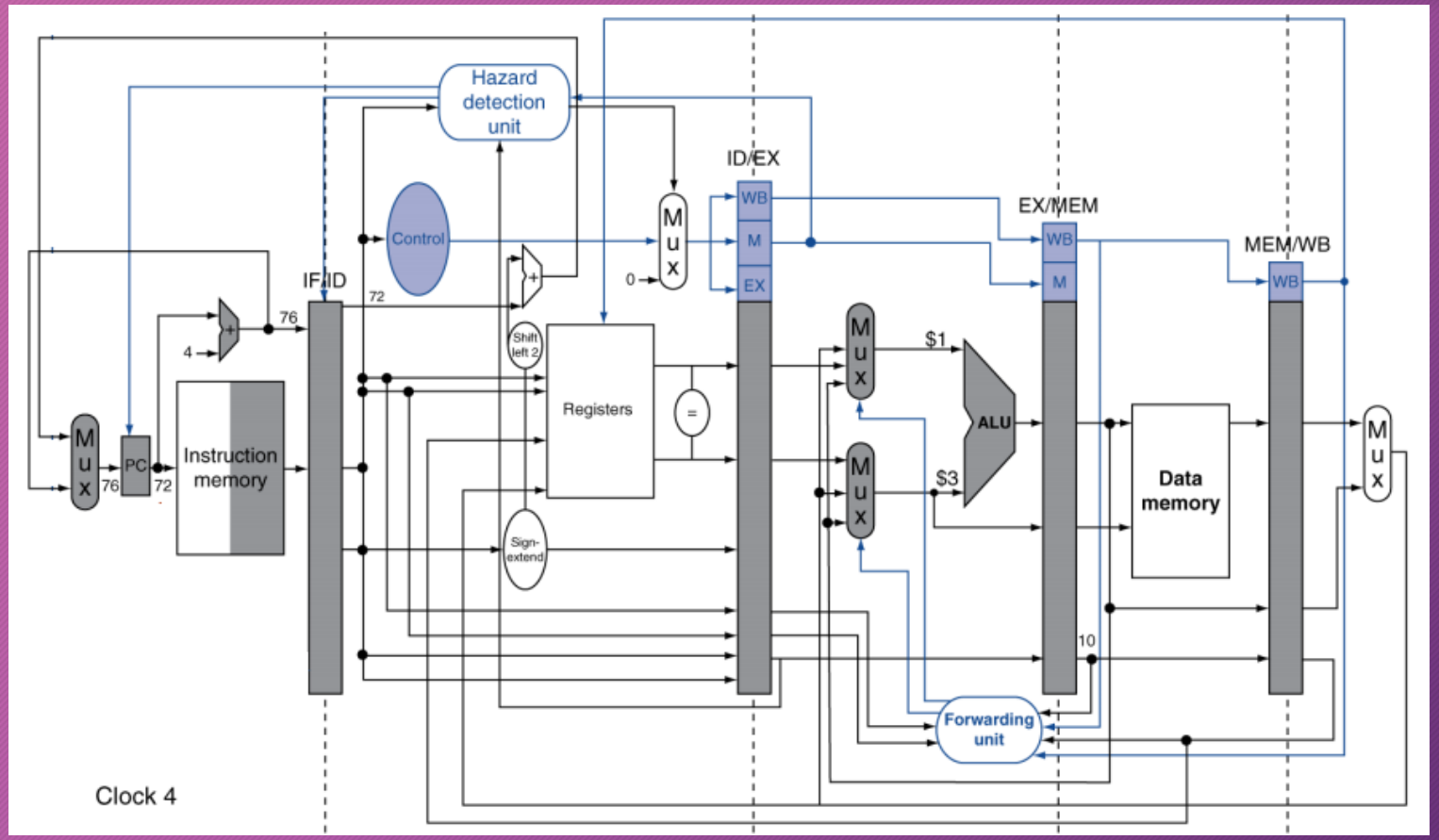


معماری کامپیوتر

استاد: دکتر محمدرضا بینش مروستی

صالح جعفریزاده – ۹۲۲۱۳۰۶۱

شمای پردازنده‌ی پیاده سازی شده



دستور های پیاده سازی شده

- R-Type
- Beg
- Lw
- Sw
- Hazardهایی که ممکن است برای این دستورات اتفاق بیفتد:
  - Control Hazard
    - که در این پروژه با استفاده از hazard Detection Unit و وارد کردن Stall به سیستم حل شده
  - Data Hazard
    - که در این پروژه با استفاده از Forwarding Unit و اضافه کردن Data Path ها و Controller های مناسب حل شده

ماژول های برنامه

- ماژول های زیر مستقیما از کد MIPS Single Cycle برداشته شدند و در ادامه از توضیح آنها چشم پوشی می کنیم
  - RegisterBank
  - InstructionMemory
  - DataMemory
  - ControlUnit
  - ALUUnit
  - ALUControlUnit
- ماژول های جدید این فاز
  - Forwarding
  - HazardUnit

ForwardingUnit

- این ماژول وظیفه ی برسی Hazard هایی که در آن $rs یا $rt دستوری ، $rd یک یا ۲دستور قبل از آن می‌باشد را بر عهده دارد

```
1  `timescale  1 ns / 1 ns
2  module forwarding_unit (
3                  input [4:0] rsAdr,
4                  input [4:0] rdAdr,
5                  input [4:0] EXMEMrt,
6                  input [4:0] MEMWBrt,
7                  input MEMWBwbRegWrite,
8                  input EXMEMwbRegWrite,
9                  output [1:0] input1Selector,
10                 output [1:0] input2Selector
11                 );
12 assign input1Selector=(MEMWBwbRegWrite == 1'b1 &&
13                 MEMWBrt != 5'b00000 &&
14                 MEMWBrt == rsAdr &&
15                 !(EXMEMwbRegWrite == 1'b1 &&
16                 EXMEMrt != 5'b00000 &&
17                 EXMEMrt == rsAdr))? 2'b01:
18                 (EXMEMwbRegWrite == 1'b1 &&
19                 EXMEMrt != 5'b00000 &&
20                 EXMEMrt == rsAdr)? 2'b10:
21                 2'b0;
22
```

```
23 assign input2Selector=(MEMWBwbRegWrite == 1'b1 &&
24                 MEMWBrt != 5'b00000 &&
25                 MEMWBrt == rdAdr &&
26                 !(EXMEMwbRegWrite == 1'b1 &&
27                 EXMEMrt != 5'b00000 &&
28                 EXMEMrt == rdAdr))? 2'b01:
29                 (EXMEMwbRegWrite == 1'b1 &&
30                 EXMEMrt != 5'b00000 &&
31                 EXMEMrt == rdAdr)? 2'b10:
32                 2'b0;
33
34 endmodule
```

Hazard Detection Unit

```
 timescale 1 ns / 1 ns
module hazard_detection_unit(
            input [31:0]instruction,
            output reg hazardDetected,
            output reg IFIDUpadte,
            output reg pcWrite
            );

always @(instruction)
    begin
    if(instruction[31:26]==6'b000100)
        begin
        hazardDetected=1'b1;
        IFIDUpadte=1'b0;
        pcWrite=1'b0;
        end
    end
```

- با توجه به دستورات پیاده سازی شده در این پروژه و وجود Forwarding Unit تنها دستور Beq میتواند باعث ایجاد Hazard در پروژه شود که این ماژول با بررسی instruction در صورتی که از نوع Beq بود Stall وارد سیستم می کند.

Core

- ماژول اصلی برنامه است که طبق شمای پروژه در اسلاید اول پیاده‌سازی شده است
- در برنامه نویسی این ماژول سعی شده از نام های مناسب برای متغیرها استفاده شود تا خوانایی کد بالا رود
  - برای مثال : IFIDIntruction نام رجیستر موجود در IF/ID است که مقدار دستور را در خود ذخیره میکند و یا EXMEMmMemRead نام رجیستر موجود در EX/MEM است که در قسمت رجیستر های مخصوص Memory(m) پرچم مربوط به خواندن حافظه را نگه می دارد
  - ماژول های اصلی هرقسمت با کامنت های یک خطی مشخص شده اند
    - برای مثال تکه کدی که زیر "////////////////////////////////MEM////////////////////" آمده مربوط به بخش Memory می‌شود
  - برای استفاده از mux3:1 از دستور زیر استفاده شده است

```
assign wireName= (selector == 2’b00)? val1:
                 (selector == 2’b01)? val2:val3;
```

نکات پایانی

- برنامه به ازای دستورات و hazard ها تست شده است.
- حافظه از ۳۲ خانه ی ۳۲ بیتی تشکیل شده است.
- آدرس دهی فایل Prog.asm بدین صورت می‌باشد
  - "E:\\prog.asm"
  - این فایل به صورت باینری در نظر گرفته شده است.
- این پروژه هم به عنوان فاز دو و هم به عنوان فاز اختیاری تحویل داده شده‌است.

“ Computers are incredibly fast, accurate, and stupid. Human beings are incredibly slow, inaccurate, and brilliant. Together they are powerful beyond imagination. ”

Albert Einstein

با تشکر از همراهیتان!