

# مستندات فنی

این مستند به تشریح جزئیات فنی و فرضیات طراحی یک پردازنده مبتنی بر معماری RISC-V می‌پردازد. طراحی حاضر یک پیاده‌سازی چند چرخه‌ای (Multi-Cycle) از مجموعه دستورالعمل‌های پایه صحیح (RV32I) به همراه افزونه ضرب و تقسیم (M) است.

## ۱. فرضیات اصلی طراحی (Design Assumptions)

طراحی این پردازنده بر پایه فرضیات زیر استوار است:

- معماری: پردازنده از نوع RV32IM است؛ یعنی یک معماری ۳۲ بیتی RISC-V با مجموعه دستورالعمل‌های پایه برای اعداد صحیح (I) و افزونه استاندارد برای عملیات ضرب و تقسیم (M).
- حافظه: حافظه به صورت بایت-آدرس‌پذیر (Byte-Addressable) در نظر گرفته شده است. طول هر دستورالعمل ۴ بایت (۳۲ بیت) است و دستورالعمل‌ها در حافظه به صورت تراز شده (Aligned) قرار دارند.
- مدل پیاده‌سازی: از پیاده‌سازی چند چرخه‌ای (Multi-Cycle) استفاده شده است. این بدین معناست که هر دستورالعمل بسته به پیچیدگی آن، در چند چرخه ساعت (Clock Cycle) اجرا می‌شود. این رویکرد در مقابل پیاده‌سازی تک چرخه‌ای (Single-Cycle) قرار می‌گیرد و امکان استفاده بهینه‌تر از سخت‌افزار را فراهم می‌کند. وجود مراحل T2، T3 و T4 در دستورالعمل‌های دسترسی به حافظه (مانند lw)، گواه این مدل است.
- واحدها: پردازنده شامل واحدهای اصلی زیر است که در دیاگرام بلوکی نیز نمایش داده شده‌اند:

- شمارنده برنامه (PC): آدرس دستورالعمل بعدی را نگهداری می‌کند.
- واحد حافظه (Memory): دستورالعمل‌ها و داده‌ها را ذخیره می‌کند.
- بانک ثبات (Register File): شامل ۳۲ ثبات عمومی (x0 تا x31) است.
- واحد منطق و حساب (ALU): عملیات حسابی و منطقی را انجام می‌دهد.
- واحد کنترل (Control Unit): سیگنال‌های کنترلی لازم برای سایر واحدها را بر اساس دستورالعمل فعلی تولید می‌کند.
- ثبات‌های کمکی: ثبات دستورالعمل (IR)، ثبات آدرس حافظه (MAR) و ثبات داده حافظه (MDR) برای تسهیل ارتباط با حافظه استفاده می‌شوند.

## ۲. دیاگرام بلوکی و مسیر داده (Datapath)

دیاگرام ارائه‌شده یک نمای کلی از مسیر داده را نشان می‌دهد. بر اساس این دیاگرام و مشخصات RTL، جریان داده در پردازنده به شرح زیر است:

- واکنشی دستورالعمل (Instruction Fetch):

1. مقدار PC در ثبات MAR قرار می‌گیرد.
2. حافظه با استفاده از آدرس موجود در MAR، دستورالعمل را خوانده و در MDR قرار می‌دهد.
3. محتوای MDR به ثبات دستورالعمل (IR) منتقل می‌شود. این مرحله در RTL با  $IR \leftarrow M[PC]$  نمایش داده می‌شود. در چرخه بعدی، PC به اندازه ۴ واحد افزایش می‌یابد.

## - اجرای دستورالعمل (Instruction Execute):

- دستورالعمل‌های نوع R (مانند add, sub):

1. مقادیر ثبات‌های منبع (rs1 و rs2) از بانک ثبات خوانده شده و به ورودی‌های ALU فرستاده می‌شوند.
2. واحد کنترل، ALU را برای انجام عملیات مورد نظر (مثلاً جمع) تنظیم می‌کند.
3. نتیجه حاصل از ALU در ثبات مقصد (rd) در بانک ثبات نوشته می‌شود. این عملیات در یک چرخه ساعت (T2) پس از واکنشی انجام می‌شود.

- دستورالعمل‌های نوع I (مانند addi, lw):

1. مقدار ثبات rs1 و مقدار فوری (immediate) از دستورالعمل به ALU فرستاده می‌شوند.
2. برای دستور addi، نتیجه جمع در ثبات rd نوشته می‌شود.
3. برای دستور lw (Load Word)، نتیجه ALU که آدرس حافظه است، در MAR قرار می‌گیرد  $(MAR \leftarrow Reg[rs1] + imm)$ . سپس داده از حافظه خوانده شده (در MDR) و در نهایت در ثبات rd نوشته می‌شود. این فرآیند به ۴ چرخه ساعت نیاز دارد.

- دستورالعمل‌های نوع S (مانند sw):

1. آدرس حافظه با جمع  $Reg[rs1]$  و مقدار فوری محاسبه شده و در MAR قرار می‌گیرد.
2. داده‌ای که باید ذخیره شود  $(Reg[rs2])$  در MDR قرار می‌گیرد (گرچه در RTL مستقیماً نوشته شده) و سپس در آدرس مشخص شده توسط MAR در حافظه نوشته می‌شود.

- دستورالعمل‌های نوع B (مانند beq):

1. مقادیر  $Reg[rs1]$  و  $Reg[rs2]$  مقایسه می‌شوند.

2. بسته به نتیجه مقایسه، PC یا با مقدار 4 + PC (حالت عادی) یا با آدرس پرش (PC + imm) به روزرسانی می‌شود.