



فاز نهایی پروژه

Computer Aided Design

CAP17 Processor

نویسنده : سهیل رستگار

در این پروژه پردازنده با معماری CISC ساخته شده است که تمامی دستورات جدول زیر را پشتیبانی می کند.

Category	Instruction	Example	Meaning	Comments
Arithmetic	add	Add d0, d1	$d0 = d0 + d1$	Two operands; overflow detected (opcode = 0000)
	add	Add d0, 25	$d0 = d0 + \text{memory}(\text{BA} + 50)$	Two operands; overflow detected (opcode = 0001)
	subtract	Sub d0, d1	$d0 = d0 - d1$	Two operands; overflow detected (opcode = 0010)
	Add immediate	Addi d0, 50	$d0 = d0 + 50$	+ constant; overflow detected (opcode = 0011)
	Multiply	Mul d0, d1	$\text{Hi, Lo} = d0 \times d1$	32 bit signed product in Hi, Lo (opcode = 0100)
Logical	And	And d0, d1	$d0 = d0 \& d1$	Two reg. operand; logical and (opcode = 0101)
	Shift left logical	Sll d0, 10	$d0 = d0 \ll 10$	Shift left by constant (opcode = 0110)
Data Transfer	Load word	Lw d0, 7	$d0 = \text{memory}(\text{BA} + 14)$	Word from memory to register ($14 = 7 \times 2$) (opcode = 0111)
	Load indirect word	Lwi d0, 7	$d0 = \text{mem}(\text{mem}(\text{BA} + 14))$	Word from memory of memory is moved to reg (opcode = 1000)
	Store word	Sw d0, 10	$\text{Memory}(\text{BA} + 20) = d0$	Word from register to memory ($20 = 10 \times 2$) (opcode = 1001)
	Store indirect word	Swi d0, 10	$\text{Mem}(\text{Mem}(\text{BA} + 20)) = d0$	Reg is store in memory of memory location (opcode = 1010)
	Clear reg or mem	CLR d0	$d0 = 0$	One operand; clear reg or clear memory (opcode = 1011)
	Move immediate	Mov BA, 50	$50 \rightarrow \text{BA}$	Move immediate to BA reg (opcode = 1100)
Compare and conditional branch	compare	CMP d0, d1	If ($d0 - d1 = 0$) Z flag = 1 Elseif ($d0 - d1 < 0$) N flag = 1 Else Z flag and N flag are zero	It does not change content of operands. CMP instruction change content of status register (flags) (opcode = 1101)
	Branch on not equal	Bne 25	$\text{PC} = \text{PC} + 2 + 50$	Bne checks Z flag and if Z flag is zero, it go to location (25×2); PC relative (opcode = 1110)
Unconditional	Simple jump	Jmp 2500	Go to 5000	Jump to target address (2500×2) (opcode = 1111)

ماژول اصلی در این پروژه به ۴ زیر ماژول حافظه اصلی، بانک رجیستر، واحد کنترل و واحد محاسبه و منطق تقسیم می شود.

حافظه ی اصلی به صورت آرایه ۸ بیتی است اما ورود و خروجی ۱۶ بیتی می گیرد. بدین صورت که در لبه بالا رونده سیگنال کلاک ۸ بیت اول و در لبه پایین رونده ۸ بیت دوم عملیات خود را (خوانده شدن، نوشته شدن) انجام می دهند. سیگنال تک بیتی Write_enable نوع عملیات را تعیین می کند.

بانک رجیستر شامل رجیستر های مورد نیاز و باس مشترک برای ارتباط بین رجیسترها می باشد.

در واحد کنترل ثبات شمارشگر دنباله، مرحله عملیاتی پردازنده را برای هر دستور (Execute , Decode , Fetch) تعیین می کند که در هر مرحله و باتوجه به دستور، سیگنال های کنترلی مورد نیاز تولید می شوند.

واحد محاسبه و منطق ، از ثبات های مشارکت کننده در عملیات های منطقی ورودی می گیرد. همینطوری دو ورودی ۴ بیتی، دو ثبات مشارکت کننده در عملیات را مشخص کرده و ورودی دیگری نوع عملیات منطقی را مشخص میکند

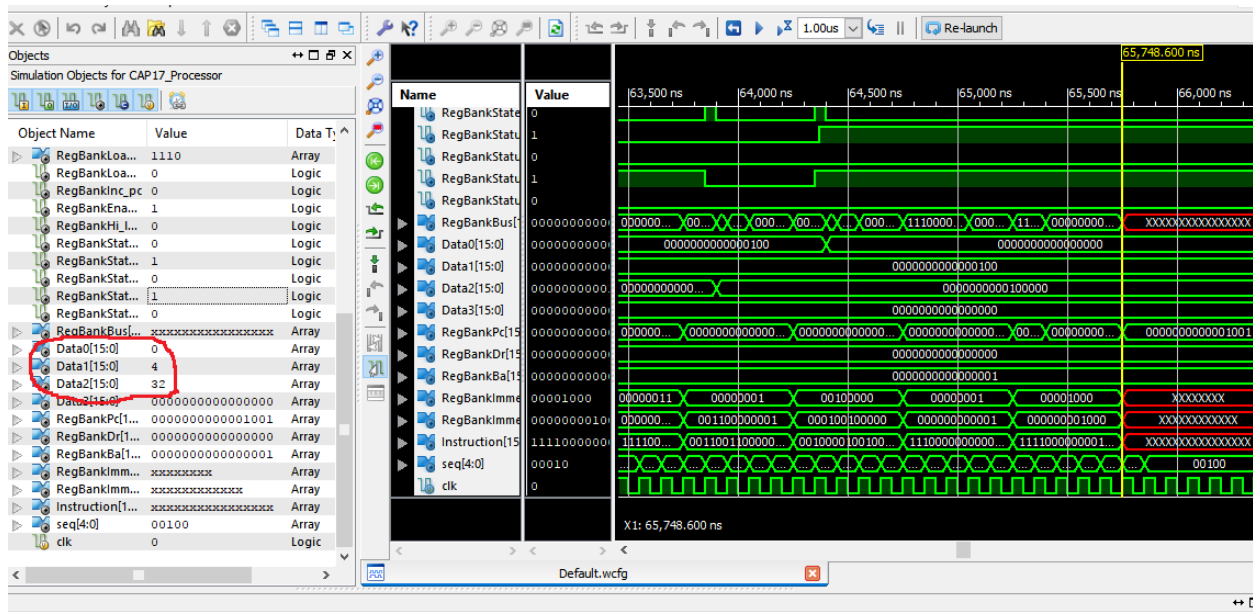
برای امتحان این پردازنده برنامه ای به زبان اسمبلی پشتیبانی شده توسط پردازنده نوشته و اجرا شد.

برنامه ی تقسیم صحیح :

```
MOV    R0,128
MOV    R1,4
CLR     R2
ADDi    R2,1      :Again
SUB     R0,R1
BNE     Loop
JMP     Out
JMP     Again     :Loop
                        :Out
```

این برنامه حاصل تقسیم صحیح عدد وارد شده در ثبات R0 را بر عدد وارد شده در ثبات R1 نشان می دهد. به دلیل استفاده از BNE عدد وارد شده در ثبات R0 حتما باید مضرب صحیح عدد وارد شده در ثبات R1 باشد

خروجی این برنامه را در شکل زیر مشاهده می کنید:



2)

xcess.

علاوه بر این برنامه، برای هر کدام از دستورات پردازنده سناریویی در نظر گرفته شده که در مازول حافظه اصلی به صورت کامنت قرارداده شده و می توان آن را اجرا کرد.

آدرس گیت هاب پروژه :

https://github.com/blu-ray/CAP17_Processor.git