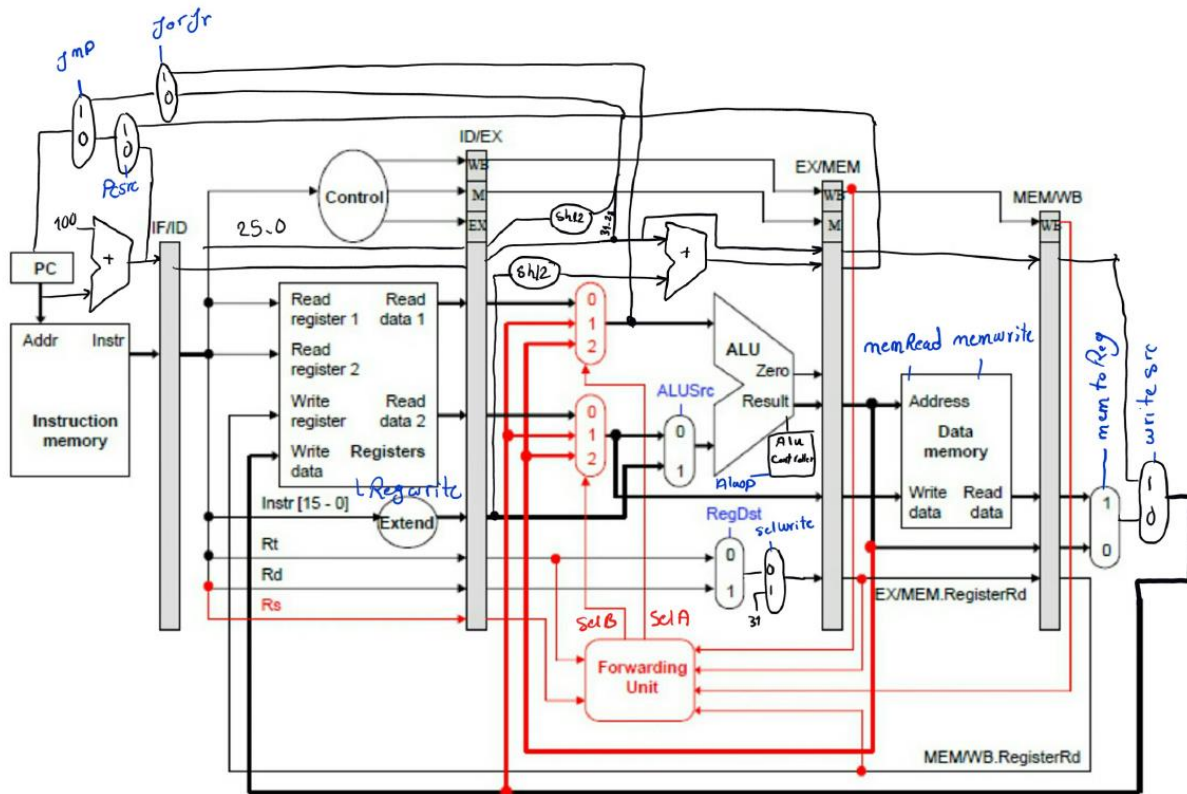
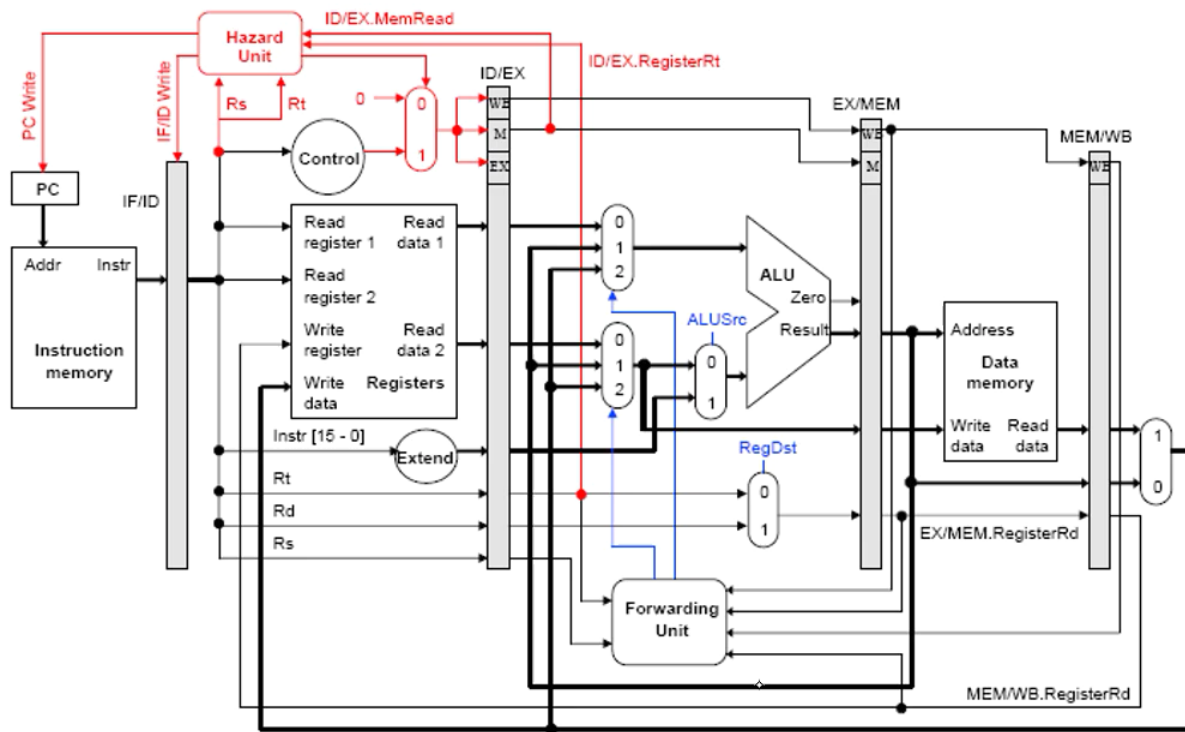


طراحی مسیر داده به شکل زیر است :



✚ برای اضافه کردن hazard unit به شکل زیر عمل میکنیم.

یک component اضافه میکنیم تا در صورتی که هزارد داشتیم ، سیگنال ها را صفر کند.



✚ طراحی کنترلر به شکل زیر است :

	RegDst	WriteSrc	WriteReg	Alusrc	branch	jmp	JorJr	AluOp	MemRead	MemWrite	MemtoReg
Slti	0	0	1	1	0	0	0	11	0	0	0
Addi	0	0	1	1	0	0	0	01	0	0	0
Jr	-	-	0	-	0	1	1	--	0	0	-
Jal	-	1	1	-	0	1	0	--	0	0	-
J	-	-	0	-	0	1	0	--	0	0	-
Beq	-	-	0	0	1	0		10	0	0	-
Lw	0	0	1	1	0	0	0	01	1	0	1
Sw	-	-	0	1	0	0	0	01	0	1	-
R-T	1	0	1	0	0	0	0	00	0	0	0

طراحی کنترلر ALU به شکل زیر است:

	OP	Func	AluOP
R-T	00	000001	000 And
		000010	001 OR
		000100	010 Add
		001000	110 Sub
		010000	111 Slt
lw/sw/addi	01	_____	010 Add
beq	10	_____	110 Sub
slti	11	_____	111 Slt

فرمت ورودی دستور ها به شکل زیر است :

Instruction format

add, sub, and, or, slt

31	26	25	21	20	16	15	11	10	6	5	0
opc		rs		rt		rd		func			

lw, sw, slti, addi

31	26	25	21	20	16	15	0
opc		rs		rt		adr or data	

J, Jal

31	26	25	0
opc		adr	

beq

31	26	25	21	20	16	15	0
opc		rs		rt		L	

Jr

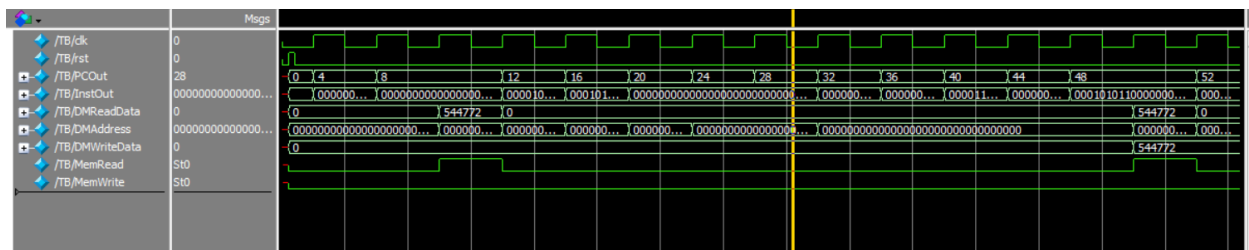
31	26	25	21
opc		rt	

برای پیدا کردن بزرگترین عنصر ارایه به شکل زیر عمل میکنیم :

```
lw R9, 1000(R0)
add R10, R9, R0
add R11, R0, R0
Loop: slti R12, R11, 20
      beq R12, R0, END
      add R14, R11, R11
      add R14, R14, R14
      lw R9, 1000(R14)
      slt R12, R9, R10
      beq R12, R0, SKIP
      add R10, R9, R0
      add R13, R11, R0
SKIP:  addi R11, R11, 1
      J Loop
END
      sw R9, 2000(R0)
      sw R13, 2004(R0)
```

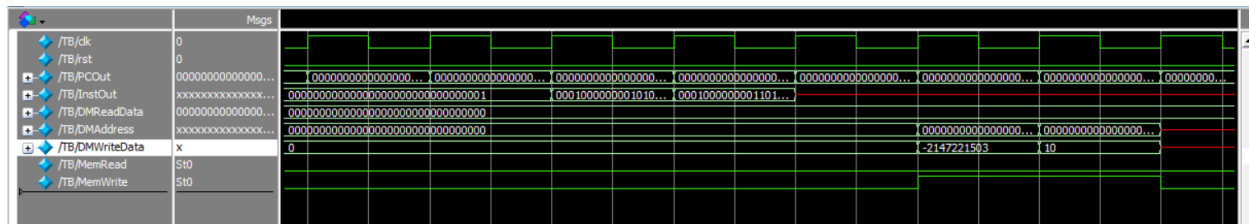
```
00001100000010010000001111101000
00000001001000000101000000000001
00000000000000000101100000000001
00001001011011000000000000010100
00010101100000000000000000010001
00000000000000000000000000000001
00000000000000000000000000000001
00000000000000000000000000000001
00000001011010110111000000000001
00000001110011100111000000000001
00001101110010010000001111101000
00000001001010100110000000010000
00010101100000000000000000000101
00000000000000000000000000000001
00000000000000000000000000000001
00000000000000000000000000000001
00000001001000000101000000000001
00000001011000000110100000000001
00000101011010110000000000000001
00011000000000000000000000000011
00000000000000000000000000000001
00000000000000000000000000000001
00010000000010100000011111010000
00010000000011010000011111010100
```

📌 ابتدای شکل موج دریافتی که میتوان در شکل زیر مشاهده می شود:



همانطور که در شکل بالا مشاهده میکنید از حافظه 1000 شروع به خواندن می کند و برای خواندن از این حافظه سیگنال MemRead را فعال میکند. سپس داده درون این بخش از حافظه استخراج می شود.

📌 انتهای شکل موج دریافتی که میتوان در شکل زیر مشاهده می شود:



حال به پایان برنامه میرسیم. میبینیم آخرین ارایه از خانه 1076 برداشته شده است و مراحل ما برای پیدا کردن بزرگترین ارایه به پایان می رسد. در نهایت کوچکترین عدد یعنی عدد 2147221503- را در حافظه 2000 مینویسیم و اندیس آن یعنی 10 را نیز در خانه 2004 حافظه مینویسیم.