

# معماری کامپیوتر

دانشکده مهندسی کامپیوتر

دکتر اسدی  
بهار ۱۴۰۳

امیرحسین صوری، ۴۰۱۱۰۶۱۸۲ و مهدی علی نژاد، ۴۰۱۱۰۶۲۶۶



## تمرین ششم عملی

سوال ۱

تست فاکتوریل

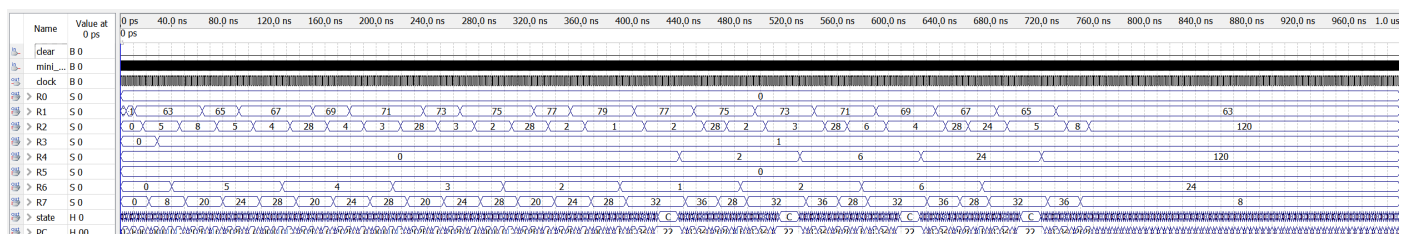
کد اسمبلی این تست به صورت زیر نوشته شده است:

		start		
0		ADDi	r1, r0, -1	r1 = 63
1		ANDi	r1, r1, -1	r1 = 63
2		ADDi	r2, r0, 5	r2 = 5
3		JAL	factoriel	do factoriel
4		BEQ	r0, r0, -1	end
5	factoriel	ADDi	r3, r0, 1	r3 == 1
6		BEQ	r2, r3, end_fact	if ( r2 == 1) end;
7		ADD	r6, r0, r2	r6 = input
8		ADD	r2, r0, r7	r2 = ra
9		JAL	push	push r2
0A		ADD	r2, r0, r6	r2 = input
0B		JAL	push	push r2
0C		SUB	r2, r6, r3	r2 = input - 1
0D		JAL	factoriel	factoriel(input - 1)
0E		ADD	r6, r2, r0	r6 = output
0F		JAL	pop	pop r2
10		MULT	r4, r2, r6	r2 = input*(input-1)
11		JAL	pop	pop r2
12		ADD	r7, r2, r0	r7 = ra
13		ADD	r2, r4, r0	r2 = output
14	end_fact	JR	r7	ret
15	push	SB	r1, r2, 0	
16		ADDi	r1, r1, 2	
17		JR	r7	
18	pop	SUBi	r1, r1, 2	
19		LB	r2, r1, 0	
1A		JR	r7	

در این کد از یک استک نرم افزاری استفاده شده است که روش کار آن در گزارش های قبلی ذکر شده است، کد به این صورت کار می کند که ابتدا مقادیر اولیه را ست کرده و سپس تابع فاکتوریل را صدا می زند، این تابع به صورت بازگشتی نوشته شده است، در اول تابع شرط پایه چک می شود که در صورت ۱ بودن ورودی، همان مقدار بازگردانده شود، در غیر این صورت بعد از ذخیره کردن دو مقدار ورودی و آدرس بازگشت در استک میجازی تابع فاکتوریل را با ورودی یکی کمتر از ورودی خودش صدا می کند، سپس جواب آن را در ورودی خودش ضرب می کند، مقدار را به صورت موقت به R۴ منتقل کرده، آدرس بازگشت را از استک پس می گیرد و حاصل را در R۲ می ریزد و باز می گرداند.

		start		
0		ADDi	r1, r0, -1	123f
1		ANDi	r1,r1,-1	527f
2		ADDi	r2, r0, 5	1405
3		JAL	factoriel	de05
4		BEQ	r0, r0, -1	003f
5	factoriel	ADDi	r3,r0, 1	1601
6		BEQ	r2, r3, end_fact	04cd
7		ADD	r6, r0, r2	f0b4
8		ADD	r2, r0, r7	f1d4
9		JAL	push	de15
0A		ADD	r2, r0, r6	f194
0B		JAL	push	de15
0C		SUB	r2, r6, r3	f796
0D		JAL	factoriel	de05
0E		ADD	r6, r2, r0	f0b4
0F		JAL	pop	de18
10		MULT	r4, r2, r6	fca3
11		JAL	pop	de18
12		ADD	r7, r2, r0	f0bc
13		ADD	r2, r4, r0	f114
14	end_fact	JR	r7	efff
15	push	SB	r1, r2, 0	8440
16		ADDi	r1, r1, 2	1242
17		JR	r7	efff
18	pop	SUBi	r1, r1, 2	3242
19		LB	r2, r1, 0	9440
1A		JR	r7	efff

این نیز کد هکس دستورات است.



و این نیز نمایی کلی از روند اجرای برنامه است، برای صحت سنجی می توانید به این مورد دقت کنید که در  $R2$  دنباله ای از ورودی ها و خروجی ها را به این صورت مشاهده می کنیم. ۵ ۴ ۳ ۲ ۱ ۲ ۶ ۲۴ ۱۲۰

در زیر نیز مرحله به مرحله ی موج با وضوح بیشتر آورده شده است برای بررسی دقیق تر عملکرد.

