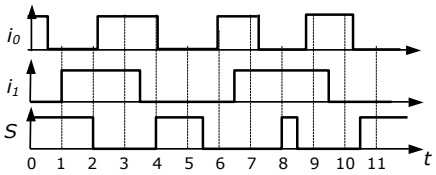
1. [2] Rappresentare il valore **+0,125** nel formato binario standard IEEE–754, singola precisione, scrivendolo come parola esadecimale.

2. [3] Scrivere la seguente funzione logica nella forma canonica più conveniente. $f(A,B,C,D) = AB + BC + D$

3. [5] Si disegni la struttura circuitale interna di un multiplexer a 2 vie e si disegni l'andamento dell'uscita in funzione degli ingressi rappresentati in figura (si trascurino i ritardi di porta).



4. [5] Si progetti un circuito caratterizzato da 4 linee di ingresso a,b,c,d e da un'uscita Y che vale '1' se e solo se la sequenza di bit <abcd> è simmetrica.
a) Determinare la tabella di verità di Y; b) esprimerla nella forma canonica più adatta; c) semplificarla mediante mappe di Karnaugh; d) semplificarla ulteriormente, se possibile, mediante semplificazioni algebriche; e) disegnarne lo schema circuitale.

5. [7] Si sintetizzi una macchina a stati finiti caratterizzata da due linee d'ingresso, IN e DIR, valutate ogni secondo, e due linee di uscita, A1 e A0, che rappresentano un numero binario di 2 bit. La macchina funziona come un contatore modulo 4: ad ogni fronte di salita sull'ingresso IN, se DIR=0 il valore in uscita viene incrementato, se DIR=1 viene decrementato. Si assumano allo stato iniziale gli ingressi e le uscite a '0'.
Si determinino: STG, STT, STT codificata e struttura circuitale del sistema completo, non trascurando la gestione del segnale di clock ed avendo cura di semplificare il più possibile le funzioni prima di tradurle in circuito.

6. [6] Si traducano in Assembly MIPS nativo (evitando di utilizzare pseudoistruzioni) le seguenti procedure in C. La procedura **MAX()** si aspetta l'indirizzo base dell'array **arr** in **$a1**, il numero di elementi **N** nel registro **$a0** e restituisce il risultato in **$v0**.

```
int MAX(int N, int arr[])
{
    int j, max = -1;
    for(j=0; j<N; j++)
        if( arr[j]>max )  max = arr[j];
    return( cubo(max) );
}
```

```
int cubo(int x)
{
  return(x*x*x)
}
```

7. [5] Rappresentare gli indirizzi ed il contenuto, byte per byte (in formato esadecimale), del segmento dati della memoria che viene modificato a seguito dell'esecuzione delle seguenti direttive (si ricorda che il codice ASCII numerico di "A"=65, "B"=66, ...):

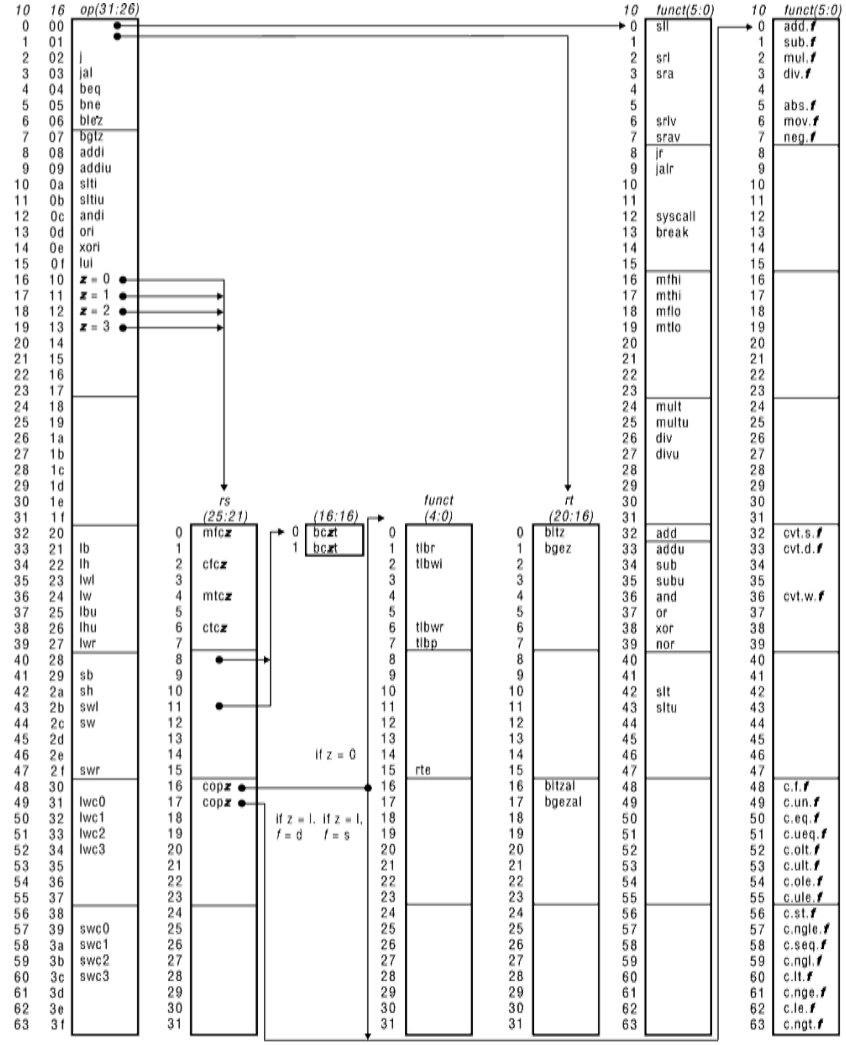
```
.data 0x300
.word -80, 0x120
.space 7
.asciiz "BICI"
.half -7
```

Se, all'inizio del programma Assembly immediatamente seguente a tali direttive, viene chiamata una system call **sbrk**, quale sarà il valore restituito nel registro **$v0**?

System calls

	codice ($v0)	argomenti	risultato
print_int	1	$a0	
print_float	2	$f12	
print_double	3	$f12	
print_string	4	$a0	
read_int	5		$v0
read_float	6		$f0
read_double	7		$f0
read_string	8	$a0,$a1	
sbrk	9	$a0	$v0
exit	10		

Registri MIPS

0	zero	24-25	t8 – t9
1	at	26-27	k0 – k1
2-3	v0 – v1	28	Gp
4-7	a0 – a3	29	Sp
8-15	t0 – t7	30	s8
16-23	s0 – s7	31	Ra

MIPS Instruction Set: