

# Tecnologie Digitali - Logbook Week 11 pt.2

Salvatore Bottaro<sup>1</sup> and Lorenzo M. Perrone<sup>2</sup>

<sup>1</sup>salvo.bottaro@hotmail.it  
<sup>2</sup>lorenzo.perrone.lmp@gmail.com

**Sommario**—Logbook di laboratorio di Tecnologie Digitali, a.a. 2015/2016. Week 11. pt.2

In figura 3 sono definiti i tempi di risposta per gli ingressi J,K e il CLOCK con S e R nello stato 0, mentre in figura 4 sono mostrati i tempi di risposta con S e R variabili.

## I. REGISTRI TIPO JK

Nel *flip-flop* di tipo JK è caratterizzato dalla presenza di due ingressi *SET-RESET* che come nel caso del *flip-flop* D prevalgono sullo stato degli altri ingressi. In più sono presenti due ingressi, J e K, anziché uno solo come nel D, che quando sono entrambi nello stato 1 abilitano lo switch delle uscite. Anche in questo caso le uscite vengono modificate solo quando nelle fasi di salita o di discesa del segnale di clock. In figura 1 è mostrata la zoccolatura del modello studiato in laboratorio, il 14027 della famiglia CMOS, mentre in figura 2 sono definiti i vari pin.

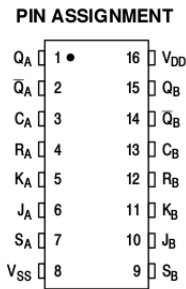


Figura 1: Zoccolatura del 14027

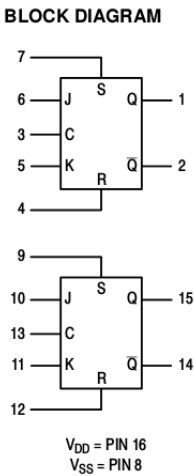
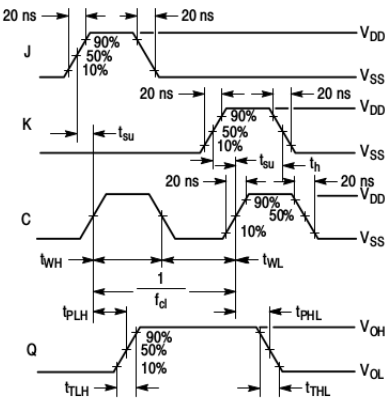


Figura 2: Definizione dei pin



Inputs R and S low.  
For the measurement of  $t_{WH}$ ,  $t_{LH}$ , and  $t_{HL}$  the inputs J and K are kept high.

Figura 3: Definizione dei tempi di risposta con S e R bassi

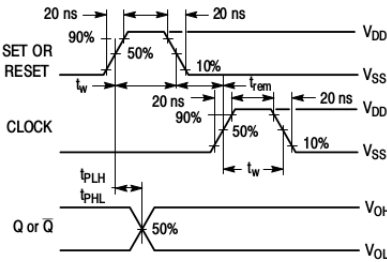


Figure 2. Dynamic Signal Waveforms (Set, Reset, Clock, and Output)

Figura 4: Definizione dei tempi di risposta con S e R variabili

In tabella I e II sono elencati rispettivamente i valori tipici di corrente, alimentazione e consumo e le fasce di tensione del 14027, del tutto identiche a quelle del *flip-flop* D.

Tabella I: Valori tipici del 14027.

Grandezza	Valore
Tensione di alimentazione	$V_{min} = -0.5V$ , $V_{max} = 18V$
Input current	10 mA
Consumo massimo	500 mW

In figura 5 è mostrata la tavola di verità del JK. Si può notare come per S e R non entrambi nulli e per segnale di

Tabella II: Fasce di tensione del 14027

Grandezza	$V_{DD}$ (V)	Valore
$V_{OL}$	5	0.05 V
	10	0.05 V
	15	0.05 V
$V_{OH}$	5	4.95 V
	10	9.95 V
	15	14.95 V
$V_{IL}$	5	1.5 V
	10	3.0 V
	15	4.0 V
$V_{IH}$	5	3.5 V
	10	7.0 V
	15	11 V

clock decrescente le uscite siano identiche al *flip-flop* D; invece per  $S=R=0$  la presenza di un ingresso aggiuntivo comporta delle differenze sostanziali, in particolare per la configurazione  $J=K=1$  con clock crescente è previsto lo switch delle uscite.

TRUTH TABLE							
Inputs						Outputs*	
$C^\dagger$	J	K	S	R	$Q_n^\ddagger$	$Q_{n+1}$	$\bar{Q}_{n+1}$
↗	1	X	0	0	0	1	0
↗	X	0	0	0	1	1	0
↗	0	X	0	0	0	0	1
↗	X	1	0	0	1	0	1
↗	1	1	0	0	$Q_0$	$\bar{Q}_0$	$Q_0$
↘	X	X	0	0	X	$Q_n$	$\bar{Q}_n$
X	X	X	1	0	X	1	0
X	X	X	0	1	X	0	1
X	X	X	1	1	X	1	1

X = Don't Care                       $^\ddagger$  = Present State  
 $^\dagger$  = Level Change                  \* = Next State

No Change

Figura 5: Tavola di verità del *flip-flop* JK

In laboratorio abbiamo verificato che il 14027 impiegato riproduce la tavola di verità di figura 5 tramite tester digitale per tutte le combinazioni possibili di S, R, J e K, in particolare abbiamo verificato che per  $J=K=1$  il *flip-flop* si comportasse come un divisore per 2.

II. DIVISORE PER 3

Come prima applicazione del *flip-flop* JK abbiamo studiato il divisore per 3 realizzando il circuito in figura 6.

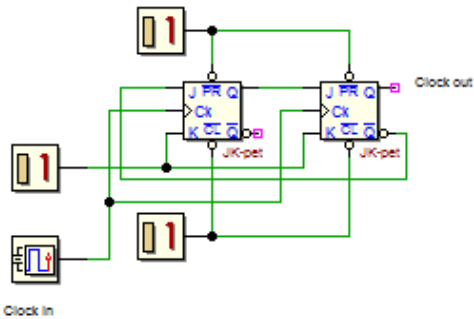


Figura 6: Schema del divisore per 3

A partire dalla tavola di verità 5 è possibile dedurre la sequenza degli stati del circuito, come si vede in tabella III.

Tabella III: Sequenza degli stati per il divisore

J1	Q1	Q2	Clock
1	0	1	↑
0	0	1	↓
0	0	0	↑
1	0	0	↓
1	1	0	↑
1	1	0	↓
1	0	1	↑

Come si nota la prima e l'ultima riga coincidono, per cui si ottiene un ciclo. Il periodo è tre volte quello di clock, per cui a livello di tavola di verità il circuito si comporta come un divisore per 3 e ci si aspetta un duty cycle di 1/3. Sempre dalla tabella III si vede come lo stato in cui  $Q1 = Q2 = 1$  non si realizza mai, per cui esso rappresenta uno stato non utilizzato. Inoltre è evidente che non esistono stati non raggiungibili, infatti il circuito possiede due gradi di libertà iniziali, ovvero due ingressi che possono essere settati liberamente, e che possono essere rappresentati da J1 e Q1. Pertanto esistono 4 stati iniziali corrispondenti alle coppie (1,1), (0,0), (1,0), (0,1), di cui i primi 3 effettivamente si riscontrano, mentre l'ultimo sparisce quando il clock si trova nello "stato" up, in quanto Q1 assume valore 0 facendo entrare lo stato del circuito dentro il ciclo. L'evoluzione del circuito si può rappresentare graficamente come in figura 7, in cui si vede come lo stato (1,1) non compare all'interno del ciclo, rimnendo quindi non utilizzato.

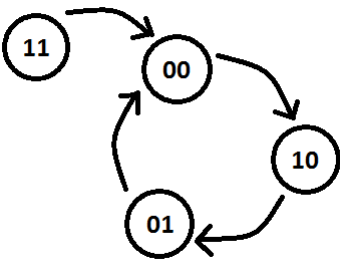


Figura 7: Stati in uscita dal divisore per 3

Abbiamo realizzato effettivamente il circuito collegando l'ingresso del clock al *digital output* 47 e impiegando il VI *Digital\_out4* per generare il segnale, abbiamo collegato le uscite Q1 e Q2 al tester digitale a abbiamo verificato che effettivamente il circuito riproducesse la sequenza di stati previsti. Abbiamo studiato la risposta del circuito collegandolo al generatore di funioni ATTEN, impostando come frequenza di clock 3 kHz. Abbiamo impiegato il fedele VI *ACQUIS\_BASE\_2015* e in figura 8 si può vedere l'esito della misura. Come si può notare il circuito si comporta ancora come divisore per 3 e anche il valore del duty cycle è quello ottenuto precedentemente.

Per verificare quantitativamente le precedenti affermazioni abbiamo impiegato i VI *Contatore\_freq* e *Contatore\_pulsew*. Tramite il primo VI abbiamo effettuato le misure dei periodi/frequenze dei segnali in ingresso

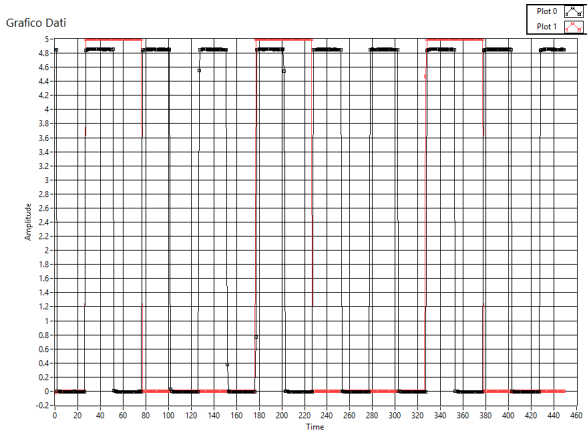


Figura 8: Confronto fra il segnale di clock (nero) e l'uscita del divisore (rosso)

e in uscita, mentre con il secondo VI abbiamo misurato il duty cycle del segnale in uscita al divisore, in entrambi i casi considerando una popolazione statistica di 100 campioni. In tabella IV e V sono mostrati i risultati relativi alle misure di frequenza in ingresso e in uscita, rispettivamente compatibili con i valori impostati e attesi di 3 kHz e 1 kHz. In tabella VI è mostrata la durata dello stato 1 sul segnale in uscita che è compatibile con  $1/3$  del periodo misurato in tabella V, per cui il duty cycle del segnale in uscita dal nostro divisore è quello previsto.

Tabella IV: Frequenze e periodi dei segnali in ingresso

Grandezza	Valore	Campioni
Frequenza	2999.86 Hz	72
	3000.29 Hz	28
Periodo	0.333301 ms	27
	0.333349 ms	73

Tabella V: Frequenze e periodi dei segnali in uscita

Grandezza	Valore	Campioni
Frequenza	999.951 Hz	23
	999.999 Hz	77
Periodo	1 ms	80
	1.00005 ms	20

Tabella VI: Misura duty cycle

Tempo	0.333351 ms	64
	0.333399 ms	36

Ovviamente non è possibile variare il segnale di clock per ottenere un segnale in uscita dal divisore con duty  $1/2$  dal momento che ciò che determina la variazione dello stato del circuito è il cambiamento dello stato del clock e non la sua permanenza in uno stato 0 o 1.

### III. SIMMETRIZZATORE

Avendo ottenuto facilmente un divisore per tre con due integrati JK, possiamo ora provare, come nella precedente esperienza, a *simmetrizzare* il segnale di uscita QB montando un circuito simmetrizzatore composto da un flip-flop D-type

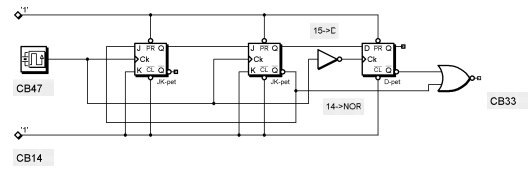


Figura 9: Schema circuitale del divisore per tre + simmetrizzatore (parte a dx della figura). Questa configurazione produce un  $\eta = 1/6$ .

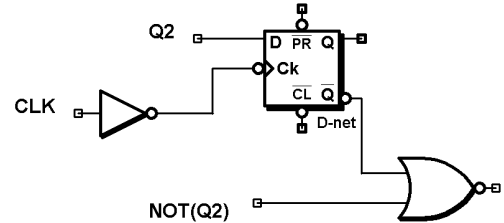


Figura 10: Circuito simmetrizzatore

e un integrato **NOR**, HC02, come mostrato in Figura (10). Contrariamente alla volta precedente, però, bisogna fare attenzione al fatto che il segnale in uscita QB da simmetrizzare ha duty cycle  $\eta = 1/3$ , e non più  $\eta = 2/3$  (in altre parole la sequenza è 001001 invece che 110110): come conseguenza di questo fatto, per ottenere in uscita un  $\eta = 1/2$  bisogna connettere QB (porta 15) con l'ingresso del **NOR** e  $\neg QB$  (porta 14) con la porta 5 (ingresso D) del flip-flop. In caso contrario ci saremmo aspettati ragionevolmente un duty di  $1/6$ . In effetti questi sono proprio i risultati che otteniamo analizzando in primo luogo con il VI ACQUIS\_BASE2 il segnale QB e l'uscita del **NOR**, rispettivamente collegati alla CB68 e alla CB33, e infine con il VI PULSE\_W collegando la CB3 separatamente ai due segnali.

Il generatore di funzioni è stato impostato ad una frequenza di 3kHz, con una tensione picco-picco  $V_{pp} = 5V$  e un offset  $V_{off} = +2.5V$ . Le porte e i collegamenti dei circuiti sono riportati in Figura (9) per il caso 15- $\zeta$ D, invece i valori ottenuti dei *duty-cycle* sono i seguenti:

	rising-time (ms)	duty-cycle $\eta$
15 $\rightarrow$ NOR, 14 $\rightarrow$ D	0.49999(5)	0.5000(1)
15 $\rightarrow$ D, 14 $\rightarrow$ NOR	0.16670(5)	0.16670(7)

Nei grafici in Figura (11, 12) sono riportate le acquisizioni di ACQUIS\_BASE2.

#### A. Hw.2

Ci chiediamo ora che tipo di segnale otterremmo se al posto del circuito **NOR** usassimo un **NAND**. Per le stesse ragioni di *simmetria* presentate nel logbook della scorsa esperienza, che discendono dal confronto della tavola di verità fra il **NOR** e il **NAND** riportata di seguito, ci aspettiamo che si invertano

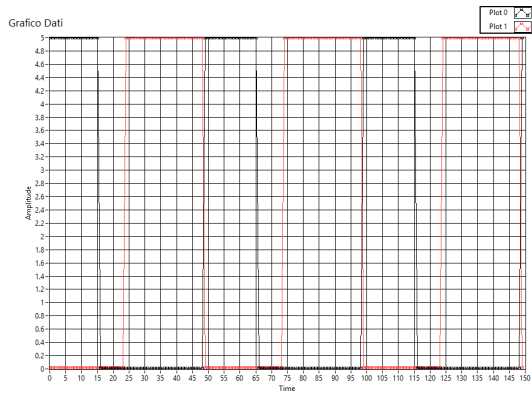


Figura 11: Segnale simmetrizzato (15- → NOR, 14- →D)

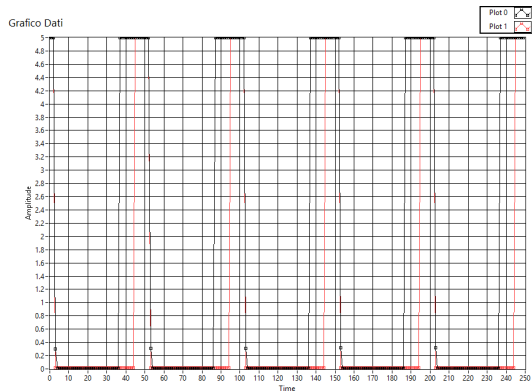


Figura 12: Segnale non simmetrizzato (14- → NOR, 15- →D), duty-cycle = 1/6

i segnali di uscita con duty di 1/2 e 1/6 in base a dove va la porta 15 QB nel circuito simmetrizzatore. Per essere più chiari: questa volta 15 →D e 14 →NOR avrà in uscita il segnale simmetrico con duty-cycle 1/2 e non più quello con  $\eta = 1/6$ .

A	B	NAND(A,B)	NOR(A,B)
1	1	0	0
1	0	1	0
0	1	1	0
0	0	1	1

Per verificarlo scriviamo la tavola di verità nel caso in cui 15- →D:

CLK	QB=D	Q	$\neg Q$	$\neg QB$	NOR( $\neg QB, \neg Q1$ )	NAND( $\neg QB, \neg Q1$ )
1	1	-	-	0	-	-
0	1	1	0	0	1	1
1	0	1	0	1	0	1
0	0	0	1	1	0	0
1	0	0	1	1	0	0
0	0	1	0	1	0	1
1	1	1	0	0	1	1

I risultati soddisfano le nostre aspettative sia nel caso 15 →D ( $\eta = 1/2$ ) che in quello (non riportato) 14 →D ( $\eta = 1/6$ ).

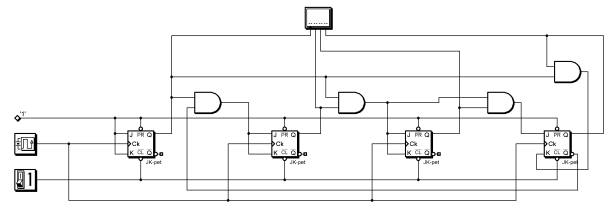


Figura 13: Schema del contatore decimale binario

#### IV. CONTATORE DECIMALE

Combinando opportunamente quattro *flip-flop* JK con dei circuiti AND, è possibile realizzare un contatore decimale a codifica binaria. Lo schema del circuito (già semplificato) è riportato in Figura (13).

Concretamente, il circuito pone alcune sfide nella realizzazione dovute al numero di circuiti da impiegare e al fatto che a nostra disposizione ci sono degli integrati NAND e non AND, motivo per cui bisognerà corocircuitare ogni volta due ingressi in più e gestire altri cavetti di cui non se ne sentiva la mancanza. Per montare con meno problemi possibile il contatore decimale, è stato impiegato uno schema realizzativo come quello in Figura (14). Organizzati in questo modo, il circuito ha funzionato al primo colpo!

Concentriamoci ora sulla sequenza degli stati del sistema, andando ad osservare le quadruplette (Q4,Q3,Q2,Q1) per ricercare gli stati irrangibili e/o inutilizzati.

q4	q3	q2	q1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
0	0	0	0

Questa è dunque la sequenza periodica degli stati (Q4,Q3,Q2,Q1), ponendo come condizione iniziale i valori (0,0,0,0). Si tratta, ovviamente, di solo 10 dei 16 stati ottenibili complessivamente. E' allo stesso tempo evidente che questi stati mancanti non vengono mai raggiunti una volta entrati nel ciclo principale (0,0,0,0) → (1,0,0,1), per cui, se vengono effettivamente raggiunti almeno una volta, ciò avverrà prima di inserirsi nella sequenza principale, e dipenderà dalle condizioni iniziali.

AVVERTENZA: per analizzare in dettaglio il comportamento di questo e della gran parte degli altri circuiti presentati nel logbook, con tutti gli stati raggiungibili, inutilizzabili (ecc...), abbiamo proceduto a simulare l'integrato JK in Python, definendo una funzione apposita JK(JK(CLK, J0, K0, Q0)) come riportata di seguito:

```

def jk(clk, j0, k0, q0):
if (clk == 1):
if (j0 != k0):
return [j0, k0, j0]
elif ((j0== 0) & (k0 == 0)):
return [j0, k0, q0]
else:
return [j0, k0, (q0+1)]else:
return [j0, k0, q0]

```

Un circuito come il contatore decimale è stato ottenuto collegando in sequenza più JK, ciascuno con le variabili in ingresso definite correttamente.

Con questi mezzi, andiamo ad osservare ora cosa succede provando ad imporre come condizioni iniziali (Q4,Q3,Q2,Q1) alcuni stati non raggiunti dal ciclo principale.

q4	q3	q2	q1
1	1	1	0
1	1	1	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
0	0	0	0

q4	q3	q2	q1
1	1	0	0
1	1	0	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
0	0	0	0

q4	q3	q2	q1
1	0	1	0
1	0	1	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
0	0	0	0

Il risultato è chiaramente visualizzabile nel grafico di Figura (15).

#### A. Hw. 3 - Divisore per 6

1) *Divisori asincroni*: I circuiti analizzati fino ad ora (e quelli che analizzeremo in seguito) possono sempre essere collegati a cascata l'uno dopo l'altro. Ad esempio è possibile

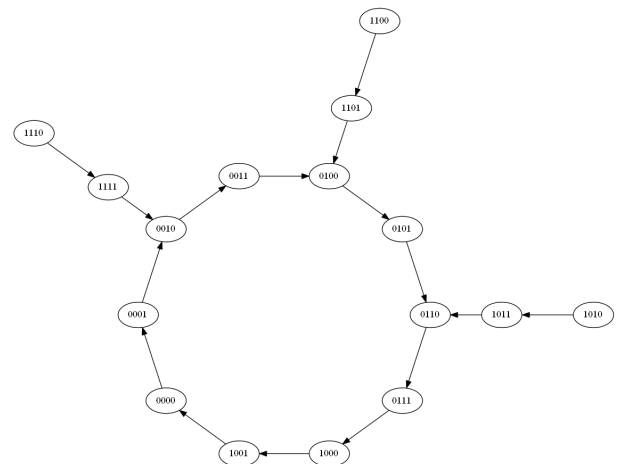


Figura 15: Grafico degli stati per il contatore decimale binario

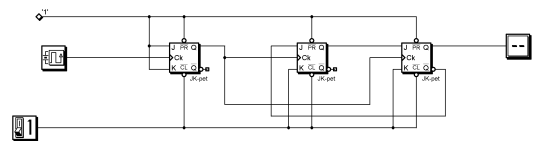


Figura 16: Divisore per sei ottenuto collegando un divisore per tre in cascata ad uno per due.

ottenere un divisore per sei collegando un divisore per tre in cascata ad uno per due (o viceversa), come mostrato in Figura (16), vale a dire collegando l'uscita a frequenza dimezzata come clock del circuito div3.

Ci aspettiamo che il duty-cycle sia  $\eta = 1/3$ , dato che il div2 è ovviamente simmetrico.

Alternativamente, è possibile collegare il divisore per due in cascata a quello per tre: la differenza principale che ci aspettiamo riguarda il duty-cycle, questa volta di  $1/2$  (Figura (17)). In Figura (18) è riportata un'immagine comparativa degli stati dei divisori asincroni ideali.

2) *Divisore sincrono*: Lo svantaggio principale dell'utilizzo di contatori asincroni sta nel ritardo nella propagazione del segnale che dall'ingresso del clock aggiorna l'output di un integrato e che deve fare da clock a sua volta di un altro integrato collegato in cascata: questa grandezza, come risulta dal datasheet del 14027, ha valori caratteristici che vanno da 25-90ns, che iniziano ad essere assolutamente rilevanti a frequenze dai MHz in su.

Per ovviare a questo problema si può provare ad ideare

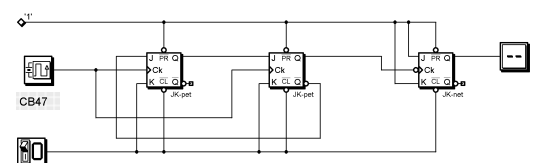


Figura 17: Divisore per sei: divisore per due in cascata ad uno per tre

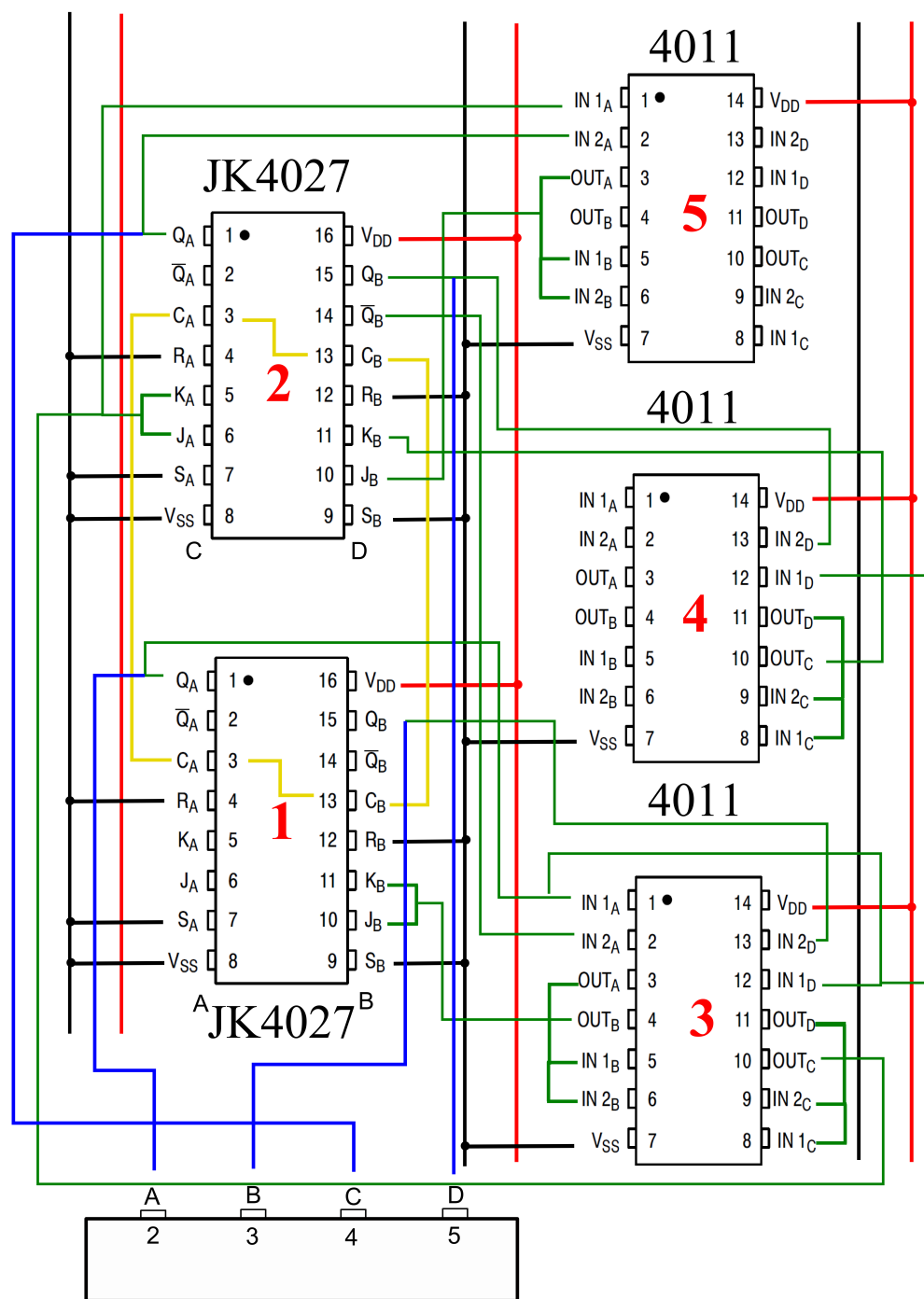


Figura 14: Schema realizzativo del contatore decimale. In giallo le linee di frequenza clock, in rosso le alimentazioni, in nero le terre, in blu le connessioni con il tester digitale e infine in verde le connessioni fra jk e NAND.

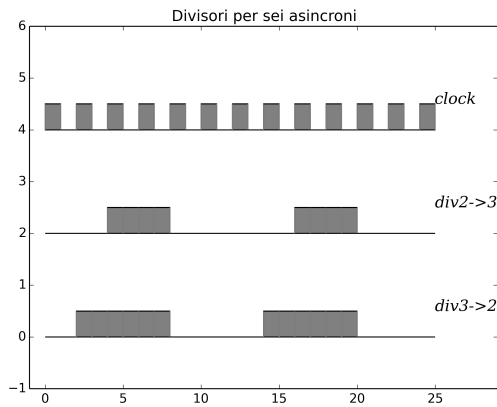


Figura 18: Divisori per sei asincroni. In ordine dall’alto in basso: segnale di clock, div3 in cascata a div2, viceversa.

j1	k1	j2	k2	j3	k3	q1	q2	q3
1	1	1	1	0	0	1	0	0
0	1	0	1	1	1	0	1	0
1	1	0	1	0	0	0	0	1
1	1	1	1	0	0	1	0	1
0	1	0	1	1	1	0	1	1
1	1	0	1	0	0	0	0	0
1	1	1	1	0	0	1	0	0

un’architettura che preveda un unico segnale di clock comune a tutti e tre gli integrati in uso. Una possibilità è quella riportata in Figura (19). Di seguito è anche riportata la tavola di verità:

B. Hm.4

Si consideri il circuito in figura 20. Calcolando la tavola di verità è possibile intuire che si tratta di un divisore per 5 con duty cycle 2/5, come si vede in tabella VII. La tabella è stata ottenuta prendendo come stato iniziale la terna (J1, K1, Q1) = (1,1,1), dal momento che tenendo conto del numero di ingressi e uscite e delle loro relazioni il circuito presenta 3 gradi di libertà.

Sempre dalla tabella si vede che durante il ciclo compaiono 5 degli 8 stati (J1, K1, Q1) possibili, pertanto gli eventuali stati irraggiungibili o inutilizzati vanno da ricercarsi nei 3 assenti. Calcolando la tavola di verità si verifica che sono tutti stati inutilizzati.

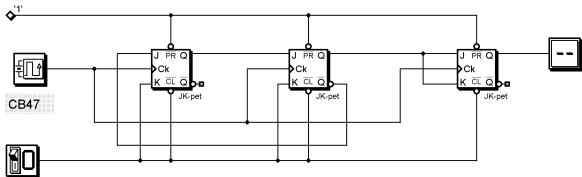


Figura 19: Divisore per sei sincrono.

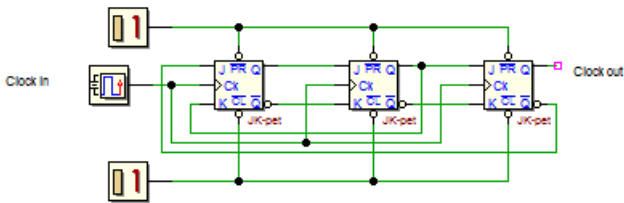


Figura 20: Schema divisore per 5

Tabella VII: Tavola del divisore per 5

J1	K1	Q1	J2	K2	Q2	J3	K3	Q3	Clock
1	1	1	1	0	1	1	0	0	↑
0	1	0	0	1	1	1	0	1	↓
0	1	0	0	1	1	1	0	1	↑
0	0	0	0	1	0	0	1	1	↓
0	0	0	0	1	0	0	1	1	↑
1	0	0	0	1	0	0	1	0	↓
1	0	0	0	1	0	0	1	0	↑
1	0	1	1	0	0	0	1	0	↓
1	0	1	1	0	0	0	1	0	↑
1	1	1	1	0	1	1	0	0	↓
1	1	1	1	0	1	1	0	0	↑
0	1	0	0	1	1	1	0	1	↓

C. Hm. 5

Si consideri il circuito in figura 21. Come si evince simulandolo con deeds o calcolandone la tavola di verità (tabella) esso rappresenta un contatore a 3 bit. Essendo un circuito con 3 gradi di libertà e dal momento che le uscite possibili sono 8, tanti quanti i possibili stati iniziali, non ci saranno stati irraggiungibili o inutilizzati.

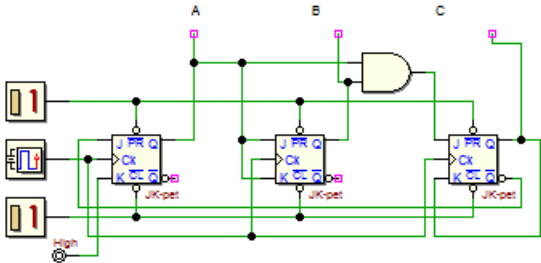


Figura 21: Schema contatore a 3 bit

Tabella VIII: Tavola del contatore per 3

J1	K1	J2	K2	J3	K3	Q3	Q2	Q1	Clock
1	1	1	1	0	0	0	0	1	↓
1	1	1	1	0	0	0	0	1	↑
1	1	0	0	0	0	0	1	0	↓
1	1	0	0	0	0	0	1	0	↑
1	1	1	1	1	0	0	1	1	↓
1	1	1	1	1	0	0	1	1	↑
0	1	0	0	0	1	1	0	0	↓
0	1	0	0	0	1	1	0	0	↑
1	1	0	0	0	0	0	0	0	↓
1	1	0	0	0	0	0	0	0	↑

La disposizione delle uscite è scelta per rendere più evidente il comportamento del circuito come contatore per 3.

D. Hm.6

Si consideri il circuito in figura 22. Se si legge la sequenza degli stati dalla tavola di verità in tabella IX, si ha che il circuito fornisce come output la sequenza, in binario, (0,1,2,5,6). Quindi non tutti gli stati entrano nel ciclo del circuito. Infatti calcolando la tavola di verità per gli stati (Q3, Q2, Q1) corrispondenti a 3, 4 e 7 si ottiene che i primi due sono stati irraggiungibili, mentre il terzo è inutilizzato. Si è usato per la tavola di verità lo stesso ordine del caso precedente.

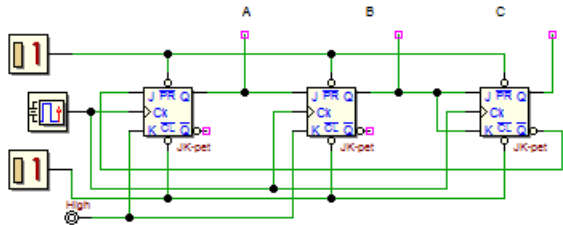


Figura 22: Schema del "contatore a salti"

Tabella IX: Tavola del "contatore a salti"

J1	K1	J2	K2	J3	K3	Q3	Q2	Q1	Clock
1	1	1	1	0	0	0	0	1	↓
1	1	1	1	0	0	0	0	1	↓
1	1	0	1	1	1	0	1	0	↓
1	1	0	1	1	1	0	1	0	↑
0	1	1	1	0	0	1	0	1	↑
0	1	1	1	0	0	1	0	1	↑
0	1	0	1	1	1	1	1	0	↓
0	1	0	1	1	1	1	1	0	↓
1	1	0	1	0	0	0	0	0	↓
1	1	0	1	0	0	0	0	0	↑
1	1	1	1	0	0	0	0	1	↓

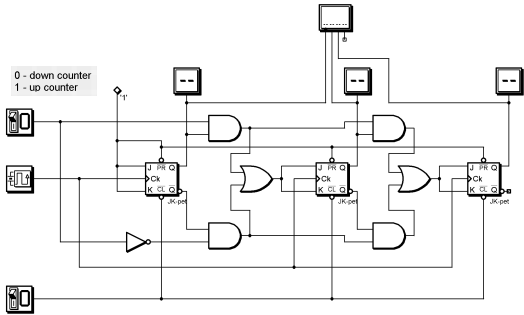


Figura 23: Contatore decimale UP-DOWN a 8 bit

Tabella X: Tavola dell'UP-counter

j1	k1	j2	k2	j3	k3	q3	q2	q1
1	1	1	1	0	0	0	0	1
1	1	0	0	0	0	0	1	0
1	1	1	1	1	1	0	1	1
1	1	0	0	0	0	1	0	0
1	1	1	1	0	0	1	0	1
1	1	0	0	0	0	1	1	0
1	1	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0

e, galvanizzati dal risultato, abbiamo deciso di aggiungere un ulteriore bit (JK) al circuito per realizzare un contatore a 16 bit (0-15). Il risultato (ben funzionante, seppur poco ordinato) è riportato nell'immagine di Figura ( ) e nello schema in Figura (24).

In entrambi i casi (8bit e 16bit), tutti gli stati sono raggiungibili (e raggiunti) circolarmente ad ogni ciclo.

E. Hw. 7

Il circuito che presentiamo di seguito è certamente il più complesso fra quelli in esame: si tratta di un contatore a codifica binaria a 3 bit (capace dunque di contare da 0-8) regolabile con un apposito interruttore in modo che possa contare anche a scendere oltre che a salire. Si tratta di un *up-down counter*. Lo schema è riportato in Figura (23). Come si vede, la struttura che lega il secondo JK al terzo JK appare a vista modulare, cioè identicamente replicabile in serie in modo da ottenere un contatore a n-bit con facilità (per lo meno in linea teorica).

Riportiamo di seguito la tavola di verità degli i/o dei tre integrati jk sia per la modalità di UP-counter che per quella di DOWN-counter. Come si vede tutte le triplette di stati (Q1,Q2,Q3) sono raggiunte ciclicamente. La realizzazione pratica è stata resa particolarmente difficoltosa dal non avere a disposizione dei NOT, OR e degli AND, per cui il numero dei circuiti NAND usati è cresciuto notevolmente. Nonostante questo, anche sperimentalmente siamo riusciti a ottenere un contatore UP-DOWN funzionante perfettamente

Tabella XI: Tavola del DOWN-counter

j1	k1	j2	k2	j3	k3	q3	q2	q1
1	1	0	0	0	0	1	1	1
1	1	1	1	0	0	1	1	0
1	1	0	0	0	0	1	0	1
1	1	1	1	1	1	1	0	0
1	1	0	0	0	0	0	1	1
1	1	1	1	0	0	0	1	0
1	1	0	0	0	0	0	0	1
1	1	1	1	1	1	0	0	0
1	1	0	0	0	0	1	1	1

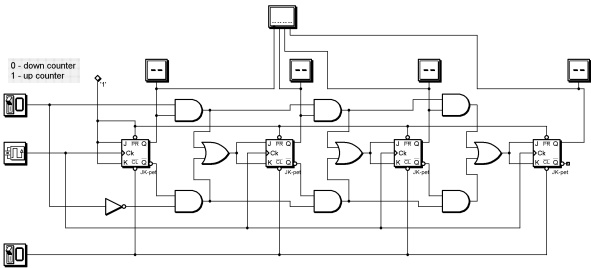


Figura 24: Contatore decimale UP-DOWN a 16 bit



Tabella XII: Tavola del contatore 4 bit

J1	J2	J3	J4	Q4	Q3	Q2	Q1
1	1	0	0	0	0	0	1
1	0	0	0	0	0	1	0
1	1	1	0	0	0	1	1
1	0	0	0	0	1	0	0
1	1	0	0	0	1	0	1
1	0	1	0	0	1	1	0
1	1	0	1	0	1	1	1
1	0	0	0	1	0	0	0
1	1	1	0	1	0	0	1
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	1
1	0	1	0	1	1	0	0
1	1	0	0	1	1	0	1
1	0	0	0	1	1	1	0
1	1	1	1	1	1	1	0
1	0	0	0	0	0	0	0

### F. Hm. 8

SI consideri il circuito in figura 25. Esso possiede 4 gradi di libertà e quindi ammette 16 possibili stati iniziali del tipo (Q1,Q2,Q3,Q4). In tabella XII è mostrata la tavola di verità, in cui compaiono solo i J e gli stati uscenti dal clock  $\uparrow$ . Come si vede il circuito si comporta come un contatore a 4 bit ed esplora tutti e 16 i possibili stati. Pertanto non si hanno stati irraggiungibili o inutilizzati.

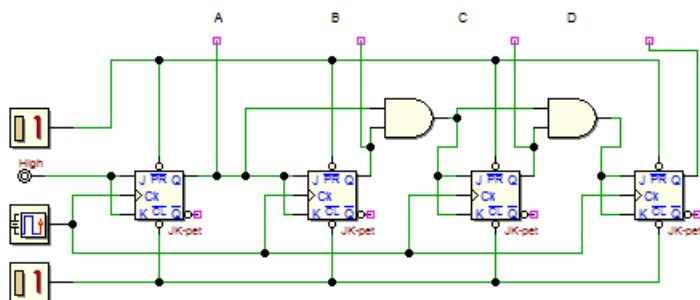


Figura 25: Schema contatore a 4 bit

### RIFERIMENTI BIBLIOGRAFICI

- [1] Product data sheet: Dual Type D Flip-Flop MC14013B. <http://onsemi.com>
- [2] Paul Horowitz, Winfield Hill - The Art of Electronics. Cambridge University Press (1989).