

Digitalstyring sammendrag

Boolsk algebra

$A + A' = 1$	$AA' = 0$
$A + A = A$	$AA = A$
$A + 0 = A$	$A \cdot 1 = A$
$A + 1 = 1$	$A \cdot 0 = 0$
$(A')' = A$	
$A + B = B + A$	$AB = BA$
$A + (B + C) = (A + B) + C$	$A(BC) = (AB)C$
$A(B + C) = AB + AC$	$A + BC = (A + B)(A + C)$
$(A + B)' = A'B'$	$(AB)' = A' + B'$

Karnaughdiagram

$$Q = A'BC'D + ABC'D' + ABC'D + BCD + ABCD + ABC$$

For hvert ledd i uttrykket setter man ett 1 tall der det gjelder. I leddene der ikke alle variablene er spesifisert setter man ett 1 tall i alle tilfellene for de uspesifiserte variablene. I resten av rutene putter man 0 tall.



	A'B'	A'B	AB	AB'
C'D'	0	0	1	0
C'D	0	1	1	0
CD	0	1	1	0
CD'	0	0	1	0






Marker ruter i rektangulær form som inneholder akkurat 2^k ruter. Du kan velge ruter som overlapper hverandre og som går fra kant til motsatt kant.

Lag nå et nytt uttrykk der du har ett ledd for hver markerte firkant. Leddet skal inneholde de variablene som er felles for alle rutene som er markert.


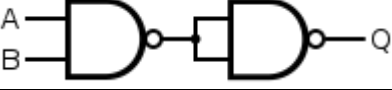
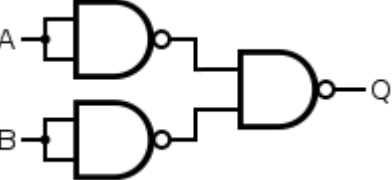
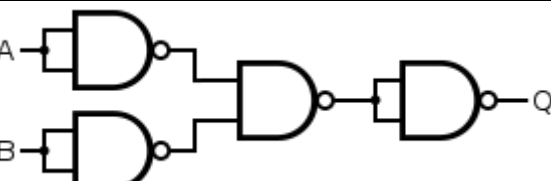
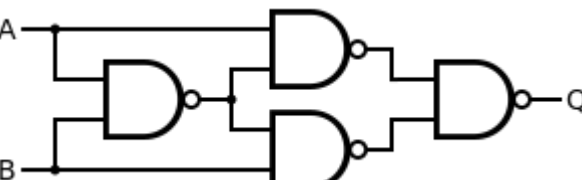
$$Q = DB + AB$$

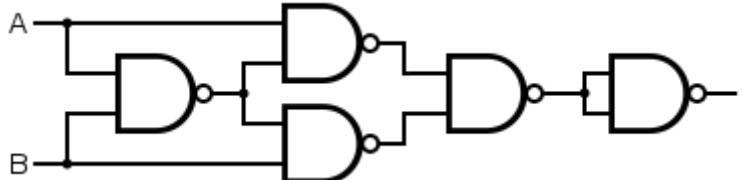
Porter

Navn	Port	Uttrykk	Sannhetstabell		
			A	B	Q
AND		$Q = AB$	0	0	0
			0	1	0
			1	0	0
			1	1	1
OR		$Q = A + B$	0	0	0
			0	1	1
			1	0	1
			1	1	1


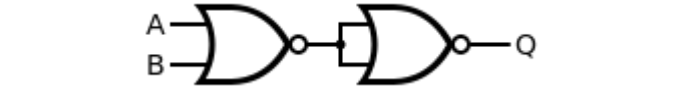
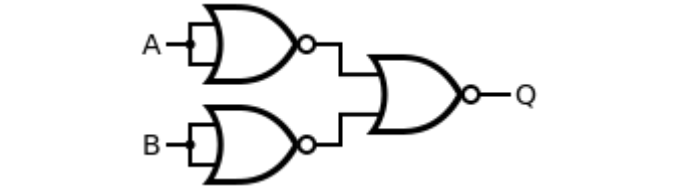
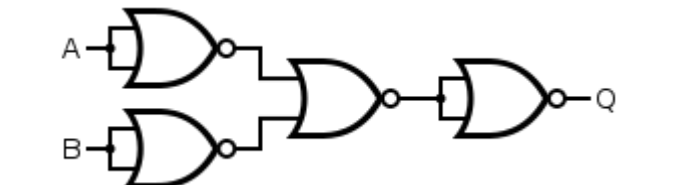
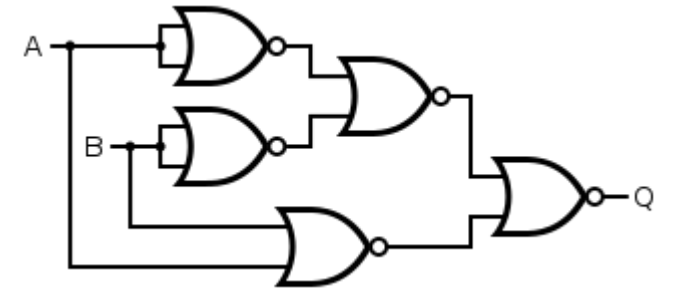
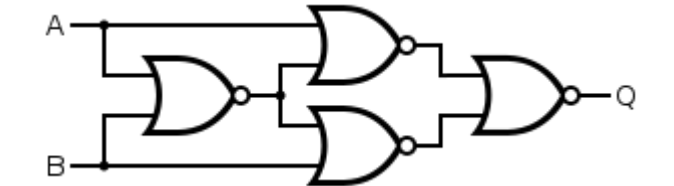
NOT		$Q = A'$	<table border="1"> <thead> <tr> <th>A</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	Q	0	1	1	0									
A	Q																	
0	1																	
1	0																	
NAND		$Q = (AB)'$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	Q	0	0	1	0	1	1	1	0	1	1	1	0
A	B	Q																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$Q = (A + B)'$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	Q	0	0	1	0	1	0	1	0	0	1	1	0
A	B	Q																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR		$Q = AB' + A'B = A \oplus B$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	Q	0	0	0	0	1	1	1	0	1	1	1	0
A	B	Q																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
XNOR		$Q = AB + (AB)' = (A \oplus B)'$	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	Q	0	0	1	0	1	0	1	0	0	1	1	1
A	B	Q																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

NAND logikk

Navn	Port	Uttrykk
NOT		$Q = (AA)'$
AND		$Q = ((AB)' * (AB)')'$
OR		$Q = ((AA)' * (BB)')'$
NOR		$Q = (((AA)' * (BB)')' * ((AA)' * (BB)'))'$
XOR		$Q = ((A(AB)')' * (B(AB)')')'$

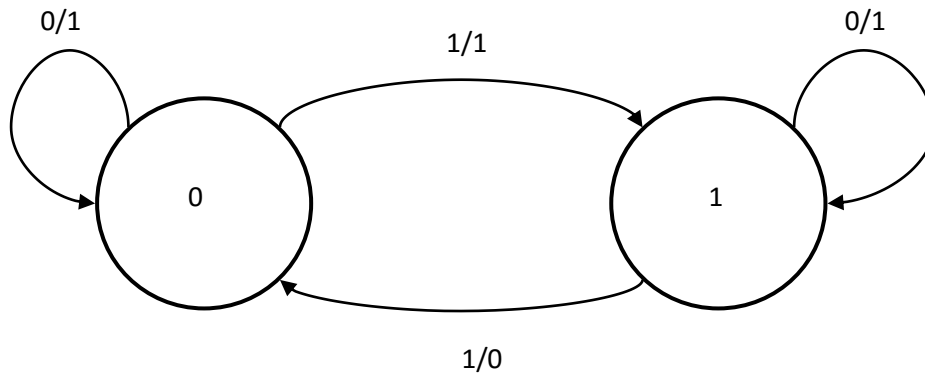
XNOR		$Q = (((A(AB))' * (B(AB))')')')$
------	--	----------------------------------

NOR logikk

Navn	Port	Uttrykk
NOT		$Q = (A + A)'$
OR		$Q = ((A + B)' + (A + B))'$
AND		$Q = ((A + A)' + (B + B))'$
NAND		$Q = (((A + A)' + (B + B))' + ((A + A)' + (B + B)))'$
XOR		$Q = (((A + A)' + (B + B))' + (A + B))'$
XNOR		$Q = ((A + (A + B))' + (B + (A + B)))'$

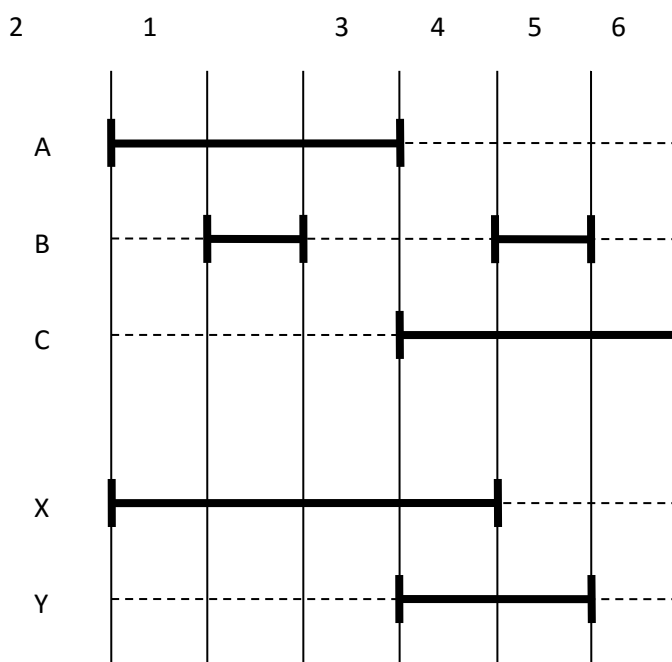
Tilstandsdiagram

Et tilstandsdiagram er et diagram som beskriver hvordan et system oppfører seg. Diagrammet har bobler som tilsvarer alle tilstandene og piler mellom de som tilsvarer alle tilstandsovergangene. Hver overgang er merket med x/y der x er input og y output i bit. Det er ikke alltid output er nødvendig.



Sekvensdiagram

Et sekvensdiagram kan ha entydige og tvetydige tilstander. Entydige tilstander er tilstander som har en unik kombinasjon av inngangssignaler. Tvetydige tilstander er tilstander som har lik kombinasjon av inngangssignaler som én eller flere andre tilstander. Det er viktig å identifisere tvetydige signaler. Hvis de tvetydige tilstandene har forskjellige kombinasjoner av utgangssignaler må man kunne skille mellom de. Man kan bruke en av utgangssignalene hvis det er kontinuerlig og er slått på ved den ene tvetydige tilstanden, og av ved den andre. Hvis ikke må man lage et ekstra minne for å skille mellom de.



I diagrammet over ser man at tilstandene 1 og 3, og 4 og 6 er tvetydige. Vi trenger ikke tenke på førstnevnte på grunn av like utgangssignaler. Sistnevnte kan løses med å bruke X til å skille de to. Lager et karnaugh diagram der man setter S der X starter og R etter X slutter. Alle tilstandene der X er høy blir Q(1), med mindre de er tvetydige med ulike utgangssignaler. Markerer Q(1), S og X i én gruppe og R og X i en annen. Q(0) kan ikke brukes i det hele tatt.

X	A'B'	A'B	AB	AB'
C'	X	X	Q(1)	S
C	Q(0)	R	X	X

$$X_{set} = A$$

$$X_{reset} = BC$$

Y	A'B'	A'B	AB	AB'
C'X'	X	X	X	X
C'X	X	X	0	0
CX	1	X	X	X
CX'	0	1	X	X

$$Y = A'B + A'X = A'(B + X)$$

Multivibratorer

En multivibrator er en elektronisk krets som kan være i 2 tilstander. Multivibratorer blir delt inn i 3 grupper.

Astabil, der kretsen ikke er stabil i noen av tilstandene. Det vil si at den aldri kan hvile i en tilstand. En typisk astabil multivibrator er en oscillator.

Monostabil, der kretsen er stabil i en av tilstandene. Det vil si at hvis du endrer tilstanden så vil den returnere til den opprinnelige tilstanden etter hvert.

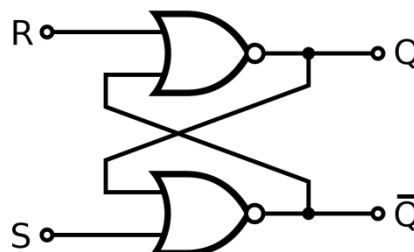
Bistabil, der kretsen er stabil i begge tilstandene. Det vil si at du alltid må bytte tilstand med et eksternt signal.

Man deler vipper inn i latch og flip-flop. Latcher sjekker for input kontinuerlig. Flip-floper ignorerer input bortsett fra når overgangen fra et dedikert klokke signal går fra lav til høy.

RS vippe

En RS flip-flop har en set og en reset inngang.

R	S	Q	Q'
0	0	0	1
0	1	1	0
1	0	0	1
1	1	Ugyldig	Ugyldig



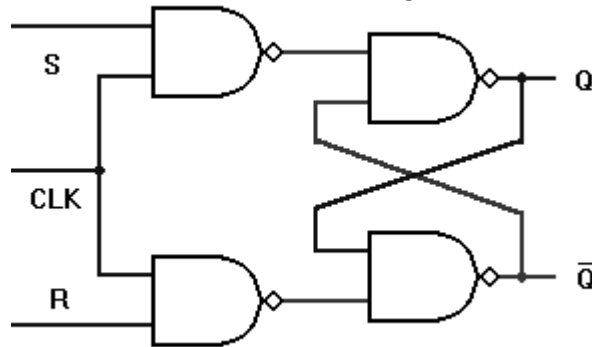
Når både R og S er satt til 1 er det ugyldig fordi da vil verdien av Q og Q' ble satt til 0 og vi vet ikke verdien av Q. Men fordi den ene banen vil være lengre enn den andre, vil den etter hvert slå seg til ro på enten 1 eller 0.

Det er enda et problem med denne vippen. Hvis vi lager en krets som tilsynelatende aldri vil kunne sette R og S til 1 på likt, som en inverter og AND port mellom de to inngangene, så vil det fortsatt være mulig at R og S blir satt til 1 på likt for en veldig kort periode på grunn av lenger bane.

Synkron RS vippe

Det tidligere nevnte problemet kan fikses ved å gjøre den avhengig av en klokke puls. Du må velge en god klokke puls, en som settes til 0 bare etter nok tid for S og R å være stabile.

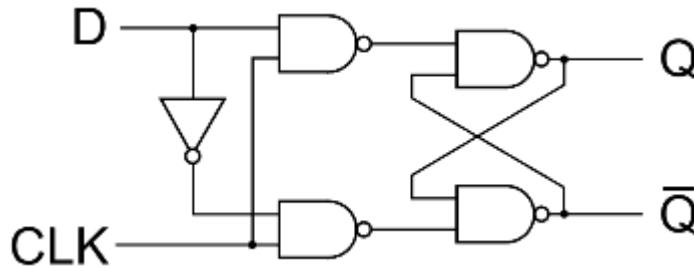
CLK	Handling
0	Ingen
1	Samme som RS



D vippe

Data vippen beholder verdien til Q mens CLK er 0 og endrer verdier til Q til verdien av D når CLK er 1.

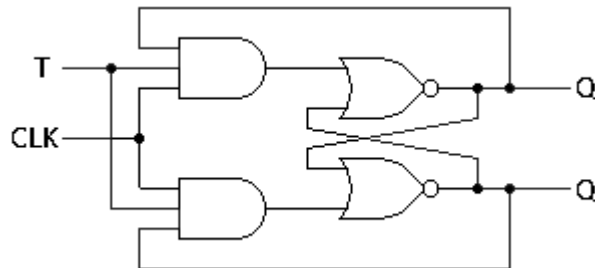
CLK	D	Q
0	X	Q
1	0	0
1	1	1



T vippe

Toggle vippen bytter tilstand når T og CLK er 1.

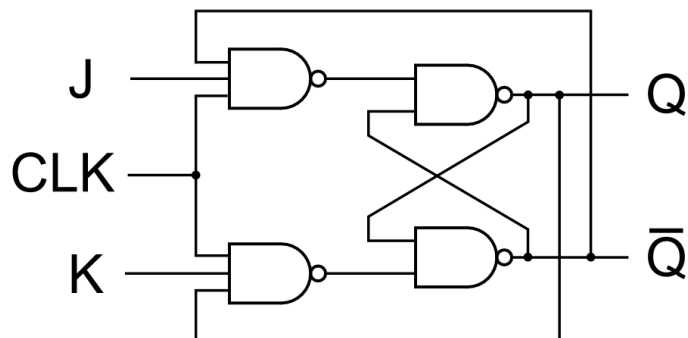
CLK	T	Q
0	X	Q
1	0	Q
1	1	Q'



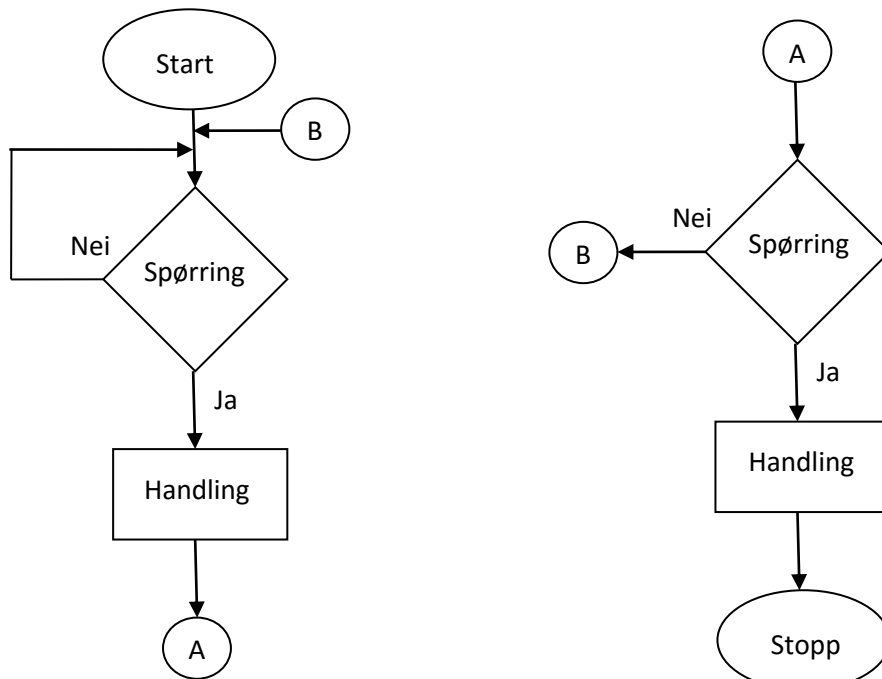
JK vippe

JK vippen fungerer som både SR og T vippe.

CLK	J	K	Q	Kommentar
0	X	X	Q	Ingen forandring
1	0	0	Q	Bevarer tilstand
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	Q'	Toggle



Flytdiagram



PLS

Datatyper

For å lagre data i minnet må du spesifisere en datatype. De forskjellige datatypene har forskjellige størrelse og kan dermed lagre forskjellige typer data. Datatypens størrelse vil bestemme hvor mange adresser den vil okkupere i minnet. For eksempel hvis du tildeler adresse 10 en double word vil den okkupere adresse 11, 12 og 13 også.

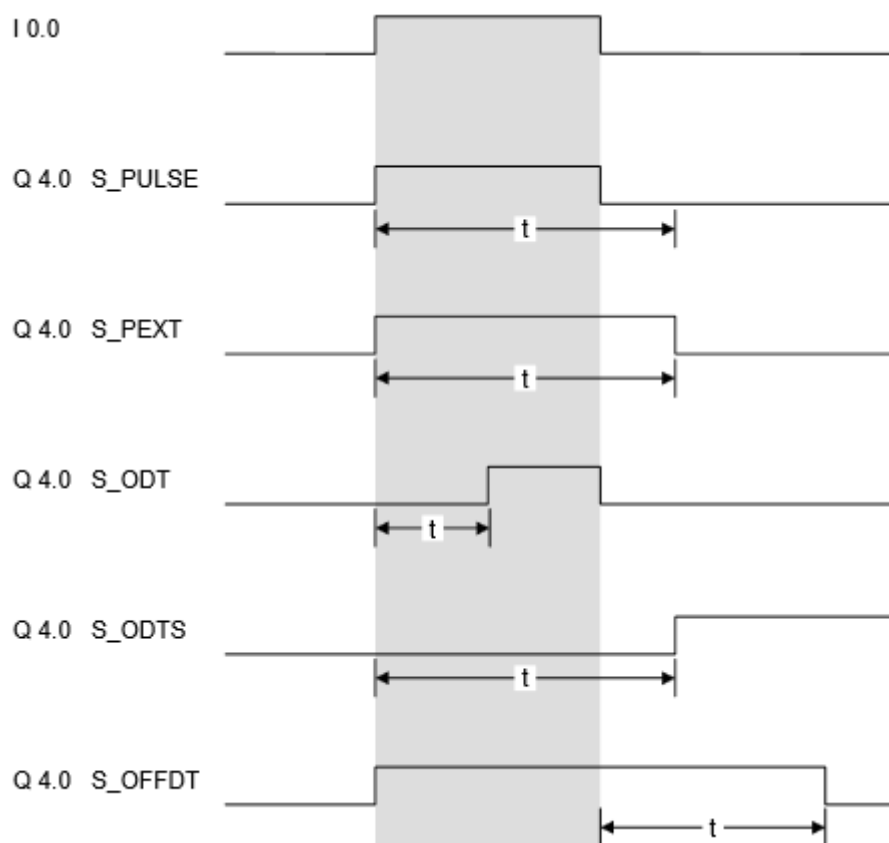
Navn	Størrelse	Eksempel adresser	Typer
Bit	1b	I0.0, Q0.1, M0.2	bool
Byte	8b, 1B ($2^8 = 256$)	IB1, QB2, MB3	byte, char
Word	16b, 2B ($2^{16} = 65536$)	IW4, QW6, MW8	WORD, DATE, INT, SSTIME
Double word	32b, 4B (2^{32})	ID9, QD13, MD17	DWORD, DINT, REAL, TIME

Konstanter

Konstanter kan defineres med [type]#[format]#[verdi], [format]#[verdi], [type]#[verdi] eller bare [verdi]. [format] kan for eksempel være 16 for hexadecimal eller 2 for binær.

Type	Eksempel
BOOL	FALSE, TRUE
BYTE	B#2#101, B#16#FF
WORD	W#16#FFFF, W#32768, W#2#1001_0100
DWORD	DW#16#FFFF_FFFF, 8#3777777777
INT	-32768, 32767, INT#16#3F_FF
REAL	123.4567, 2e4, 1.23E+02
S5TIME (SIMATIC time)	S5T#5H10S, S5T#1H_1M_1S_10MS
Counter	C#10

Timere



Timer	Beskrivelse
S_PULSE Pulse timer	Den maksimale tiden ut signalet er 1 er den samme som verdien t . Ut signalet kan ble satt til 0 fortere hvis inn signalet skifter til 0 før t .
S_PEXT Extended pulse timer	Ut signalet er 1 i tiden t , uansett om inn signalet endres til 0 før.
S_ODT On-delay timer	Ut signalet er bare 1 etter tiden t og inn signalet fortsatt er 1.
S_ODTS Retentive on-delay timer	Ut signalet går bare fra 0 til 1 etter tiden t uansett om inn signalet er 0 eller 1.
S_OFFDT Off-delay timer	Ut signalet går til 1 når inn signalet blir 1. Når inn signalet blir 0 så starter t å telle. Etter tiden t settes ut signalet til 0 igjen.

Referanser

- [1] D. A. Sahu, «Clocked R-S, D, J-K and T Flip-Flop,» 5 Desember 2011. [Internett]. Available: <http://www.iitg.ernet.in/asahu/cs221/Lects/Lec13.pdf>.
- [2] PLC 4 good, «Siemens S7 SCL cheat sheet,» 23 Oktober 2013. [Internett]. Available: http://plc4good.org.ua/files/03_downloads/SCL_table/SCL-cheat-sheet.pdf.
- [3] T. K. Wroldsen, *MAS218 forelesninger*, 2014.