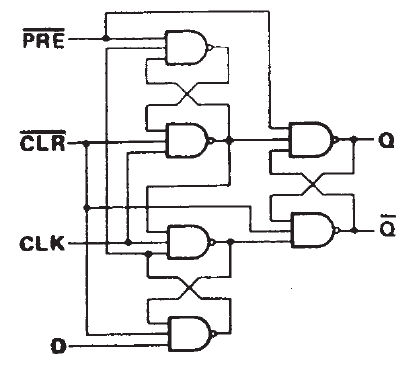
PLD D type flip flop with asynchronous set reset

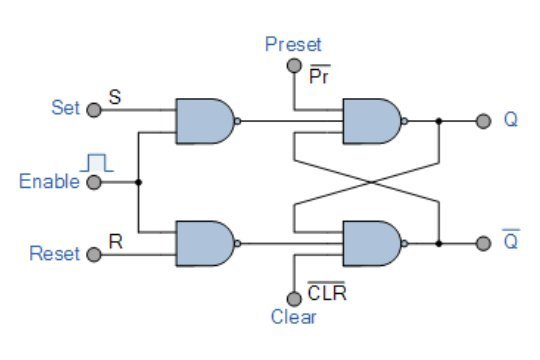
Questo tipo di circuito viene utilizzato per gestire alcune logiche del PC.

# D type flip flop with asynchronous set reset

From SN74LS74 ( Texas Instruments ) datasheet



# D type flip flop with synchronous set reset



L’ingresso CLEAR ( CLR ) serve per assicurarsi che l'uscita Q venga sovrascritta con un livello logico 0 ( BASSO ).

L’ingresso PRESET ( PRE ) serve per assicurarsi che l’uscita Q venga sovrascritta con un livello logico 1 ( ALTO ).

Se uno o entrambe gli ingressi CLEAR o PRESET, si trovano a livello logico 0 ( BASSO ), l’ingresso CLOCK non viene considerato. In questi stati di CLEAR e/o PRESET il livello logico delle uscite non cambia mai.

In qualsiasi momento l’ingresso CLEAR viene portato a livello logico 0, non viene considerato lo stato dell’ingresso D. L’uscita Q viene portata a livello logico 0 ( BASSO ), l’uscita !Q viene portata a livello logico 1 ( ALTO ).

In qualsiasi momento l’ingresso PRESET viene portato a livello logico 0, non viene considerato lo stato dell’ingresso D. L’uscita Q viene portata a livello logico 1 ( ALTO ), l’uscita !Q viene portata a livello logico 0 ( BASSO ).

Se entrambe gli ingressi CLEAR e PRESET si tronano a livello logico 1 ( ALTO ), un **fronte di salita** sull’ingresso CLOCK, copia lo stato dell’ingresso D sull’uscita Q.

L’uscita !Q è sempre l’inverso ( complementare ) dell’uscita Q.



PER UNA DESCRIZIONE PIU’ DETTAGLIATA DEL CHIP 74LS74 e DEI RELATIVI FLIP FLOP IN ESSO CONTENUTI VEDERE IL RELATIVO DATASHEET.

PLD Per ricreare il Flip Flop

Solo alcune PLD supportano alcune funzionalità necessarie per ricreare facilmente la logica del Flip Flop. Dal manuale di WinCupl ( doc0737.pdf è possibile verificare quale serie di PLD si presta meglio alla realizzazione de

WinCupl EXTENSIONS ( Vedere il documento doc0737.pdf )

.D D input of D-type flip-flop

.R R input of SR-type output flip-flop

.S S input of SR-type output flip-flop

.SP Synchronous preset of flip-flop

.AP Asynchronous preset of flip-flop

.SR Synchronous reset of flip-flop

.AR Asynchronous reset of flip-flop

**.SP** Extension

The .SP extension is used to set the Synchronous Preset of a register to an expression. For example. the equation:

Y.SP = A & B ; /\* A and B are inputs \*/

causes the output Y to be preset synchronous with the local clock used in the macro cell when inputs A and B are true.

**This feature is supported on Atmel ATF22V10B, ATV750/B and ATV2500/B devices which share Synchronous preset product terms**.

**.AP** Extension

The .AP extension is used to set the Asynchronous Preset of a register to an expression. For example. the equation "Y.AP = A&B;" causes the register to be asynchro

nously preset when A and B are logically true.

**This feature is supported on the Atmel ATF1500 family of devices.**

**.OE** Extension

The .OE extension is used to specify a product term driven output enable signal. It is required for using bi-directional I/O and the individually programmable output-enable

product terms available on Atmel ATV750/B, ATV2500/B and ATF1500 family of devices.

**Atmel Devices which have a pin-controlled OE inputs, such as the ATF1500 family also use this suffix.**

**.D** Extension

The .D extension is used to specify the D input to a D-type register. This options causes the compiler to configure the macrocells in the Atmel PLD device to D-type reg isters.

**For Atmel PLD’s such as the ATF16V8B/20V8B/22V10B, ATV750/B and ATV2500/B the .D extension must be used for registered logic. Otherwise, CUPL will generate an error.**

**.AR** Extension

The .AR extension is used to define the expression for Asynchronous Reset for a reg ister. This is used in devices that have one or more product terms connected to the

Asynchronous reset of the register.

**Devices which have a pin-controlled reset inputs, such as the Atmel ATF1500 family also use this suffix.**

Name Flip\_Flop;

PartNo 00;

Date 04/03/2025;

Revision 01;

Designer Engineer;

Company None;

Assembly None ;

Location ;

Device g22v10;

/\* INPUT \*/

PIN 1 = clock;

PIN 2 = data;

PIN 3 = !clr;

PIN 4 = !pre;

PIN 13 = !outen;

/\* OUTPUT \*/

PIN 14 = Q1;

PIN 15 = Q2;

/\* Logic Equations \*/

Q1.OE = outen;

Q2.OE = outen;

Q1.SP = pre;

Q1.AR = clr;

Q1.D = data;

Q2 = !Q1;