Chapter 17O

Online Content: Counters

17O.1 74LS469 alternatives

The 74LS469 8-bit synchronous up/down counter is considered an obsolete part.[1] As of early 2024, it was still available from several distributors (try https://LAoE.link/Find_74LS469.html) and parts can often be found on eBay (https://LAoE.link/eBay.com).

If you are unable to find a 74LS469 or find one at a reasonable price, we offer two alternatives below.

Make one from 74LS169 counters: The 74LS169 is a 4-bit synchronous up/down counter that appears to have better availability than the 74LS469: see https://LAoE.link/Find_74LS169.html. It has almost an identical pinout as the '160 - '163 series of up counters except that $\overline{\text{CLR}}$ is replaced by U/$\overline{\text{D}}$. (However, unlike the '16X counters, the carry in and out are active low.) It is fairly easy to wire two 74LS169 counters (and optionally a 74HCT541 or 74LS541 3-state buffer) to emulate a 74LS469: see Fig. 17O.1.[2]

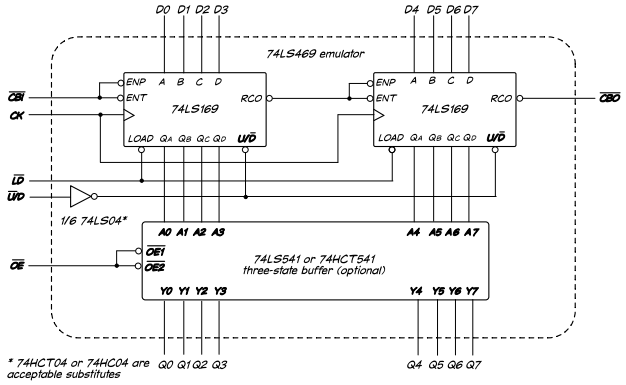

Figure 17O.1: 74LS469 emulation with two 74LS169 counters.

[1] One hint an LS part may be considered obsolete is if there is no HC equivalent.

[2] The circuit will work fine with a 74HC04 but the input levels on the U/$\overline{\text{D}}$ input will be CMOS not TTL.

The 74HCT541 octal 3-state buffer is optional. While the lab does suggest you might want to test the 74LS469's output enable function, it probably is not worth the extra effort to wire it in; feel to skip it. (If you do wire it in, ignore the buffer reference designations in the Fig. 17O.1 and just connect the buffers to the closest counter inputs; they are all identical.) Also, if you prefer, you can wire the ($\overline{\text{ENP}}$) enable on both the counters and one of the two octal buffer enables to ground if that is easier than wiring the respective pairs of enables together.[3]

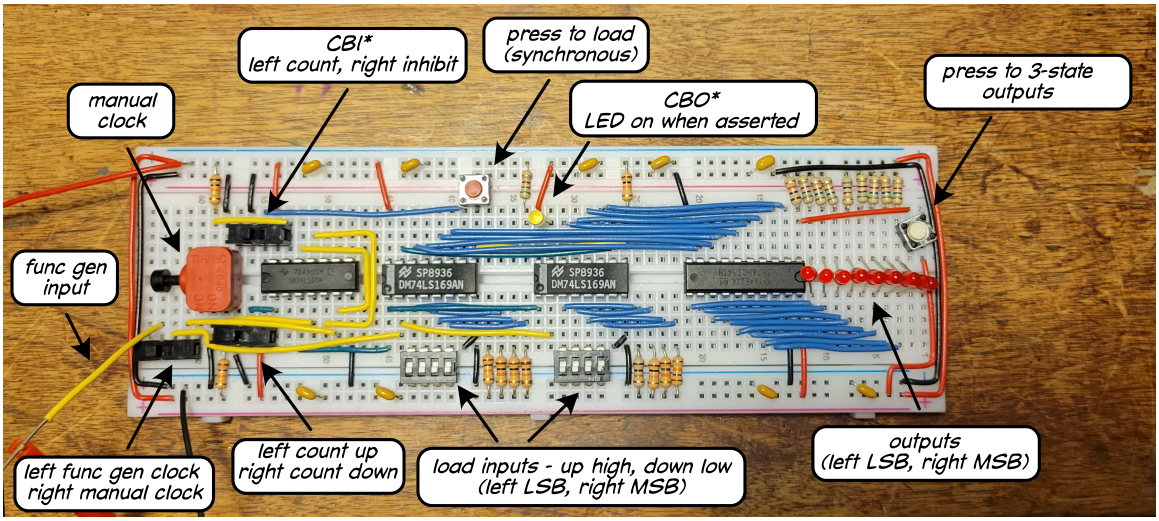

Figure 17O.2: 74LS469 emulation build with self-contained inputs and outputs.

Make one in the WebFPGA: You can always use the WebFPGA (or other PLD) to create 7400 parts. (One advantage of using the WebFPGA is that it is already wired to the breadboard LOGIC SWITCHES.) If you use the WebFPGA, all input signals must be kept to $\leq 3.3V$. Here is the Verilog file:[4]

```
//////////////////////////////////////////////////////////////////////////////////
// Company: Learning the Art of Electronics
// Engineer: David Abrams
//
// Create Date:   2020-07-15
// Module Name:   CTR_7LS469.v
// Project Name:  74LS469
// Target Device:  WebFPGA
// Description: Verilog design for a 74LS469 counter
```

[3]When cascading only two '169 counters, it does not matter which enable ($\overline{\text{ENP}}$ or $\overline{\text{ENT}}$) is connected to the RCO of the lower order counter as long as you ground the other active-low enable. However, if you cascade more than two counters, you must use $\overline{\text{ENT}}$ with the third and higher counter as only it is ANDed with the counter's Q outputs to create the counter's $\overline{\text{RCO}}$ output. $\overline{\text{ENP}}$ is the AND of the Q outputs only, so using it to enable a higher order counter would allow that counter to increment or decrement when the previous counter is at terminal count, even if lower order counters are not.

[4]Available at https://LAoE.link/FPGA/17O_74LS469.v

```
//              (but it is 3.3V CMOS, not a 5V LS part, when built with the FPGA)
////////////////////////////////////////////////////////////////////////////////

//*********************************************
// The comments in this block are directives to
// the WebFPGA environment.
//
// Set name of top level module (default is fpga_top)
// @FPGA_TOP CTR_74LS469
//
// Commands to connect the named signals below
// to specific WebFPGA output pins.
// @MAP_IO CK      30  // PB1 NC - press for rising edge
// @MAP_IO LD_bar   3  // PB2 NO - press for low
// @MAP_IO CBI_bar 27  // SW9
// @MAP_IO UD_bar  26  // SW10
// @MAP_IO OE_bar  12  // external slide switch
// @MAP_IO CBO_bar 15  // external LED
// @MAP_IO D[0]    11  // logic switch 8
// @MAP_IO D[1]    10
// @MAP_IO D[2]     9
// @MAP_IO D[3]     8
// @MAP_IO D[4]     7
// @MAP_IO D[5]     6
// @MAP_IO D[6]     5
// @MAP_IO D[7]     4  // logic switch 1
// @MAP_IO Q[0]    17  // logic indicator 8
// @MAP_IO Q[1]    18
// @MAP_IO Q[2]    20
// @MAP_IO Q[3]    21
// @MAP_IO Q[4]    22
// @MAP_IO Q[5]    23
// @MAP_IO Q[6]    24
// @MAP_IO Q[7]    25   // logic indicator 1
//*********************************************

module CTR_74LS469 (
  input CK,
  input LD_bar,
  input CBI_bar,
  input UD_bar,
  input OE_bar,
  input [7:0] D,
  output CBO_bar,
  output [7:0] Q
);

  reg [7:0] count; // internal counter for 74LS469 count

  always @(posedge CK) begin
    if (!LD_bar)
      count <= D; // synchronous load
    else if (!CBI_bar && !UD_bar) // only increment if enable asserted
      count <= count + 1;
```

```
    else if (!CBI_bar && UD_bar)
      count <= count - 1;
  end

  // output is count or high impedance if output enable not asserted
  assign Q = OE_bar ?  8'bz : count ;

  // carryout is asserted at terminal count only if carry in is also asserted (to cascade counters)
  assign CBO_bar = !(((count == 8'hff && !UD_bar) || (count == 8'h00 && UD_bar)) && !CBI_bar);

endmodule
```

Please send comments or corrections to: authors@LAoE.link