

Μετατροπείς Ψηφιακού σε Αναλογικό Digital to Analog Converter

Ο μετατροπέας Ψ/Α δέχεται ως είσοδο μια δυαδική λέξη n bits και δίδει στην έξοδο του μια τάση ανάλογη προς τον δυαδικό αριθμό που παριστάνουν τα n bit της εισόδου:

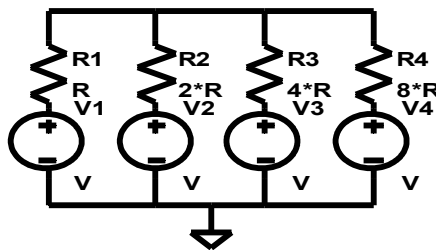
$$V_{out} = (b_{n-1} \cdot 2^{n-1} + b_{n-2} \cdot 2^{n-2} + b_{n-3} \cdot 2^{n-3} + \dots + b_0 \cdot 2^0) \cdot \frac{V_R}{2^n - 1}$$

όπου V_R μια τάση αναφοράς, που πρέπει να είναι όσο το δυνατόν πιο σταθερή. Η σταθερά στον παρανομαστή μπορεί να διαφέρει ανάλογα με την υλοποίηση.

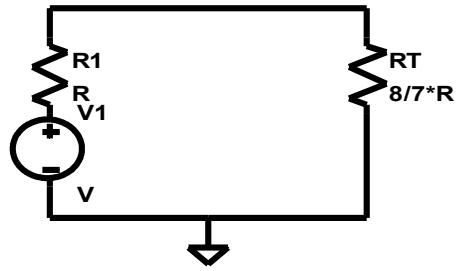
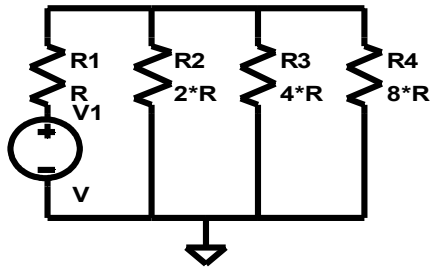
Ένας πολύ απλός μετατροπέας 4 ψηφίων φαίνεται στο ακόλουθο σχήμα. Η τάση εξόδου είναι η τάση στον κοινό κόμβο των αντιστάσεων, ενώ την ψηφιακή είσοδο απαρτίζουν οι 4 πηγές V_1, V_2, V_3, V_4 που παίρνουν είτε την τιμή 0 είτε την τιμή V_R . Η τάση αυτή δίνεται από τον τύπο:

$$V_{out} = \frac{1}{15} \cdot (V_1 \cdot 2^3 + V_2 \cdot 2^2 + V_3 \cdot 2^1 + V_4 \cdot 2^0)$$

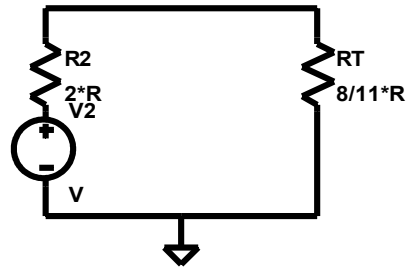
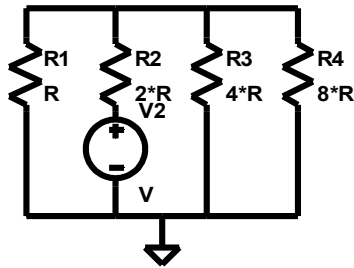
ό οποίος προκύπτει, με βάση την αρχή της επαλληλίας, αν υπολογίσουμε την τάση εξόδου λαμβάνοντας υπ' όψη μια πηγή μόνο κάθε φορά και θεωρώντας τις άλλες 0 όπως φαίνεται στα 4 σχήματα της επόμενης σελίδας. Αν θέσουμε $V_j = b_j \cdot V_R$, με b_j δυαδικά ψηφία έχουμε την σχέση εισόδου εξόδου του μετατροπέα Ψ/Α. Το πιο σημαντικό ψηφίο αντιστοιχεί στην πηγή που έχει εν σειρά την μικρότερη αντίσταση, δηλ την V_1 . Το κύκλωμα αυτό μπορεί να επεκταθεί σε περισσότερα ψηφία, επειδή όμως οι αντιστάσεις είναι όροι γεωμετρικής προόδου με λόγο 2 θα οδηγηθούμε αφ' ενός σε πολύ μεγάλες αντιστάσεις και αφ' ετέρου είναι πρακτικά δύσκολο να επιτύχουμε σταθερό λόγο 2. Υπάρχουν ειδικά ολοκληρωμένα που κάνουν την μετατροπή αυτή, όπως και την αντίστροφη, με περισσότερα ψηφία και με μεγαλύτερη ακρίβεια και σταθερότητα.



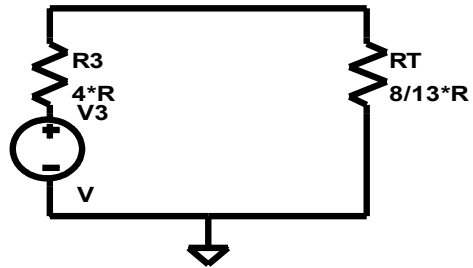
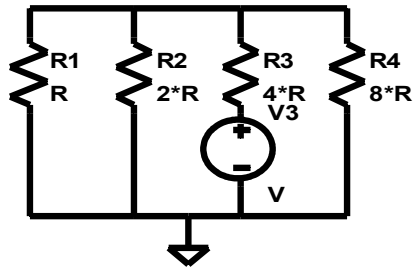
Στο πείραμα αυτό θα χρησιμοποιήσετε ως είσοδο του μετατροπέα τις εξόδους ενός δυαδικού μετρητή 4 ψηφίων και θα παρακολουθήσετε στον παλμογράφο την έξοδο του μετατροπέα. Εάν χρησιμοποιήσετε τον μετρητή 74163 ο οποίος μετράει πάντα προς τα πάνω θα παρατηρήσετε ένα σήμα που μοιάζει με σκάλα και είναι μια αρκετά καλή προσέγγιση του πριονωτού παλμού. Το σχετικό κύκλωμα φαίνεται στην σελίδα 3. Αν πάλι χρησιμοποιήσετε τον 74191 με το κύκλωμα της σελίδας 4 θα πάρετε ένα τριγωνικό παλμό. Αν η έξοδος του μετρητή οδηγούσε τις διευθύνσεις μιας μνήμης, της οποίας οι έξοδοι θα οδηγούσαν το δίκτυο αντιστάσεων, θα είχατε μια πολύ γενική γεννήτρια κυματομορφών.



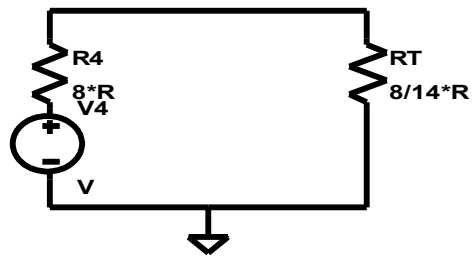
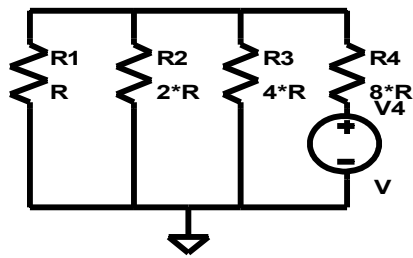
$$V_{out} = V_1 \cdot \frac{R_T}{R + R_T} = V_1 \cdot \frac{\frac{8}{7} \cdot R}{R + \frac{8}{7} \cdot R} = V_1 \cdot \frac{8}{15} = \frac{V_1}{15} \cdot 2^3$$



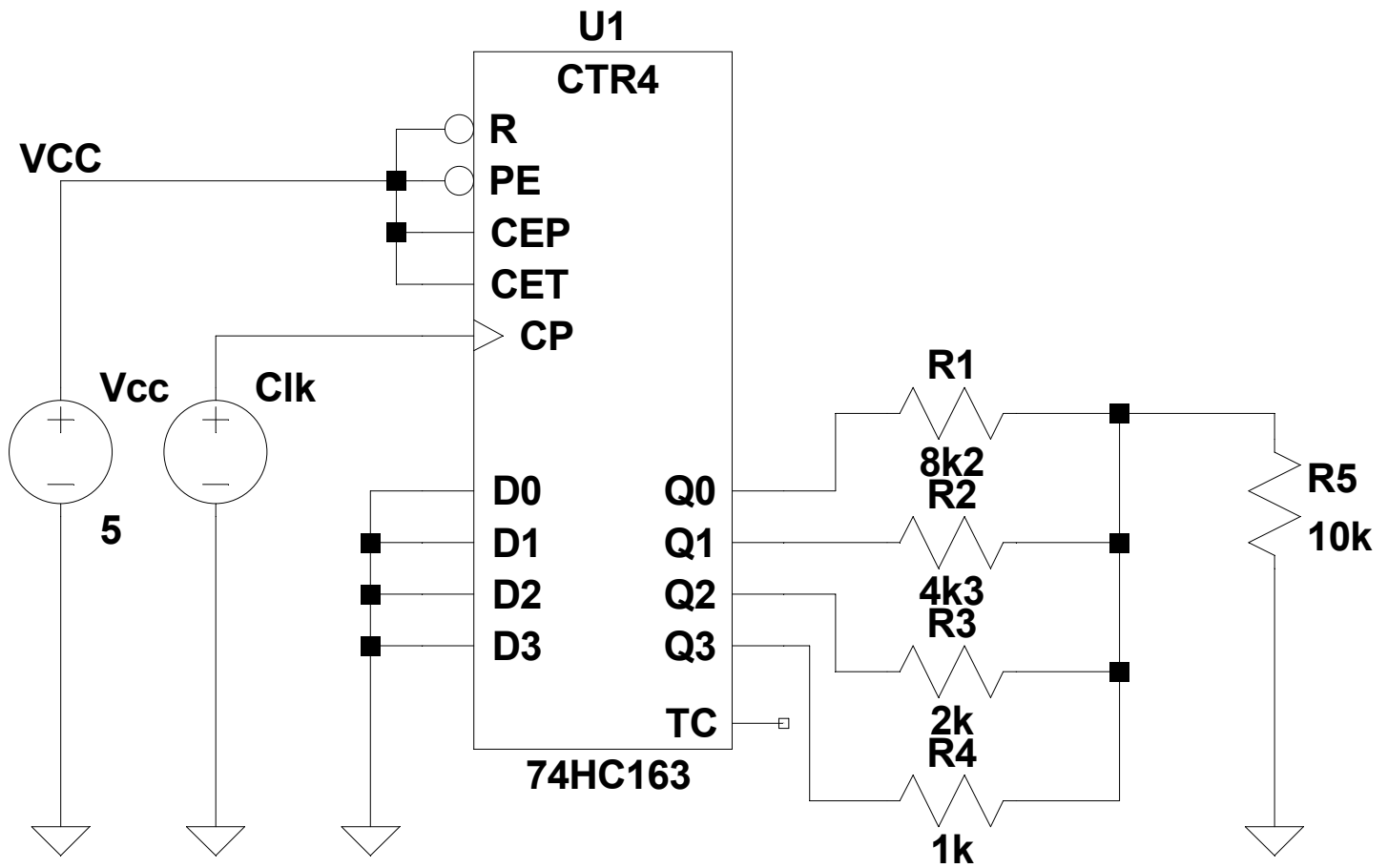
$$V_{out} = V_2 \cdot \frac{\frac{8}{11} \cdot R}{2 \cdot R + \frac{8}{11} \cdot R} = V_2 \cdot \frac{4}{15} = \frac{V_2}{15} \cdot 2^2$$



$$V_{out} = V_3 \cdot \frac{\frac{8}{13} \cdot R}{4 \cdot R + \frac{8}{13} \cdot R} = V_3 \cdot \frac{2}{15} = \frac{V_3}{15} \cdot 2^1$$



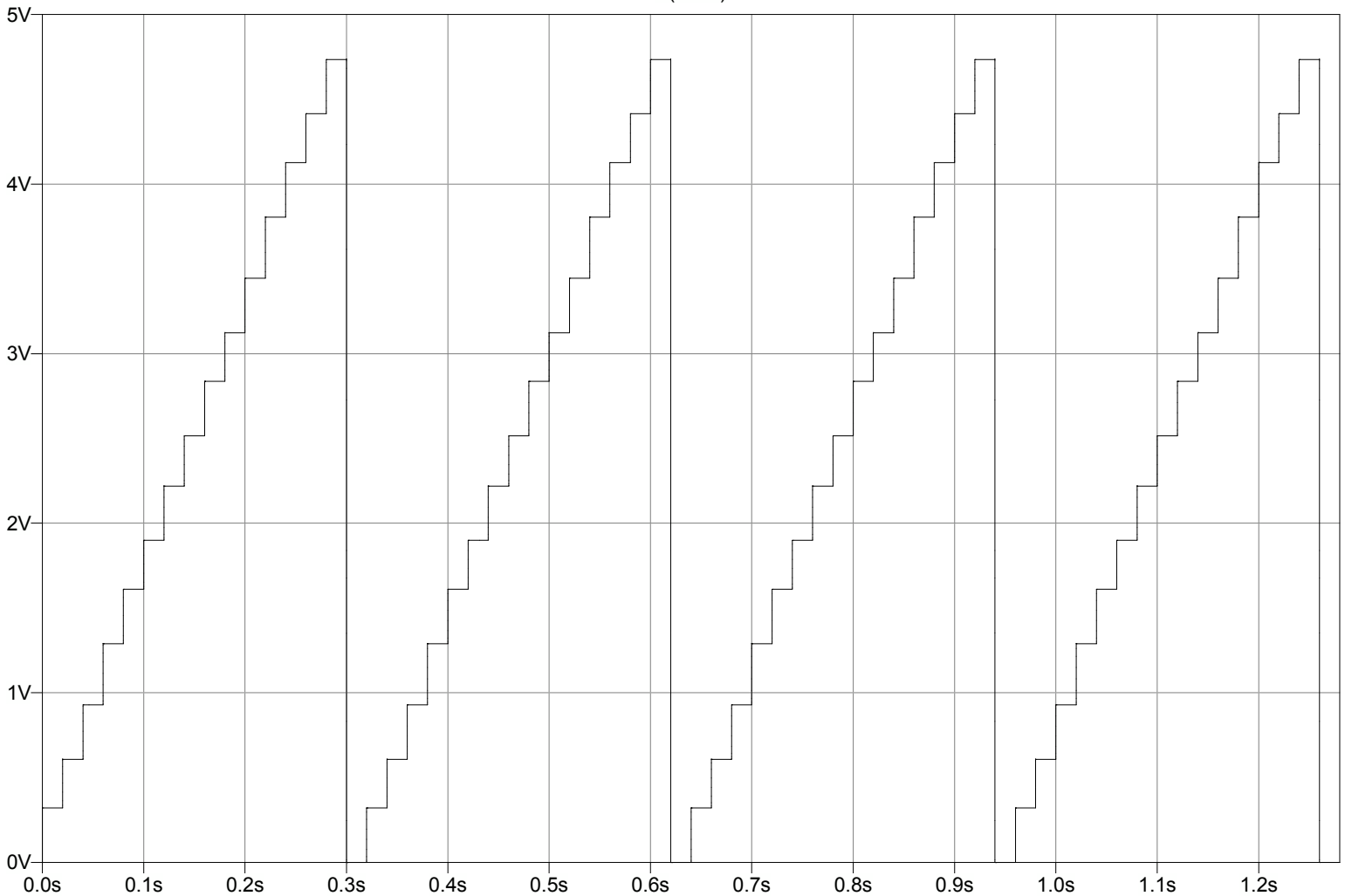
$$V_{out} = V_4 \cdot \frac{\frac{8}{14} \cdot R}{8 \cdot R + \frac{8}{14} \cdot R} = V_4 \cdot \frac{1}{15} = \frac{V_4}{15} \cdot 2^0$$



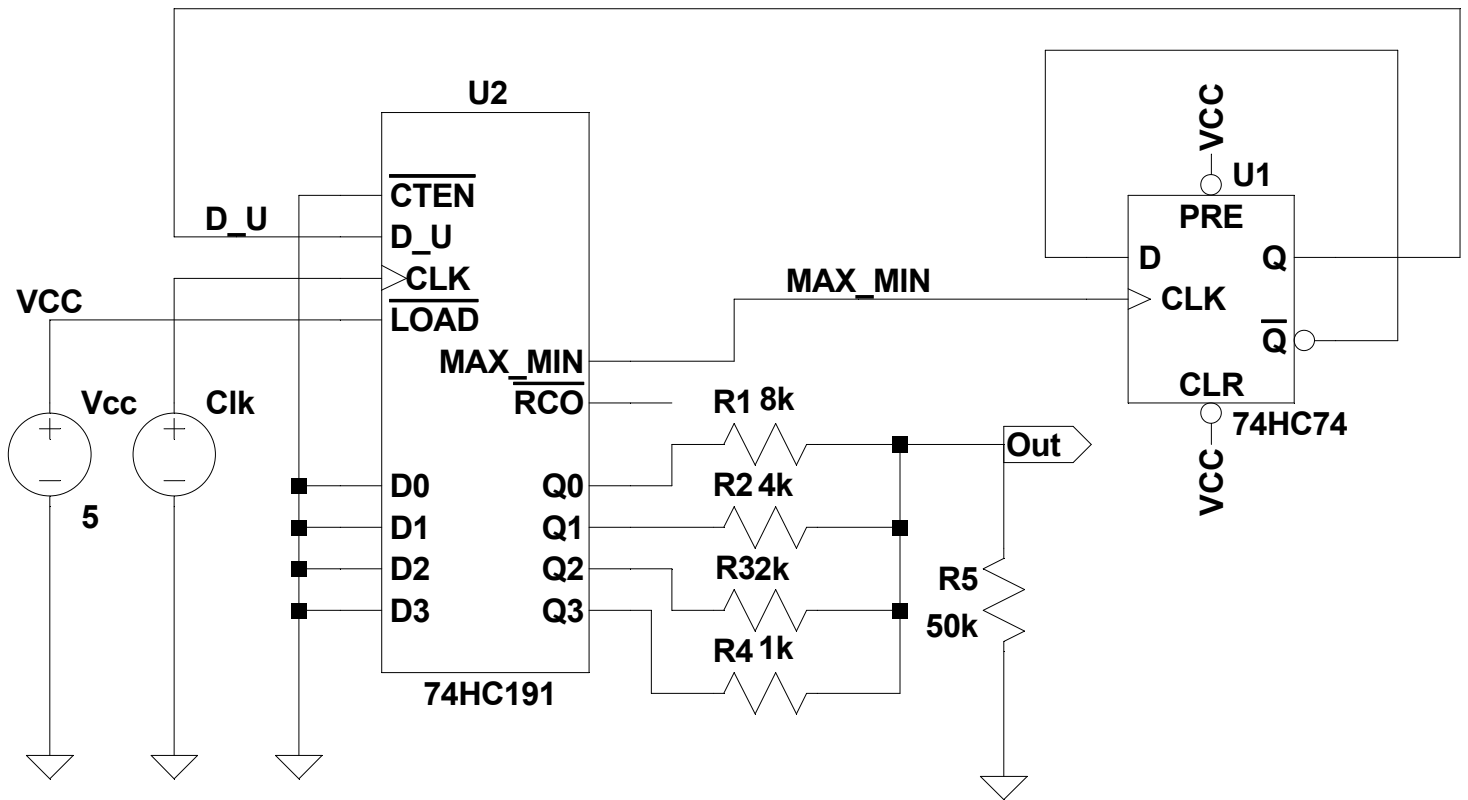
.include 74hc.lib .tran 1280ms .wave saw.wav 16 44.1k V(n001) V(n003)

--- C:\Program Files\LTC\LTspiceIV\examples\MyCircuits\DAConverter.asc ---

V(n003)



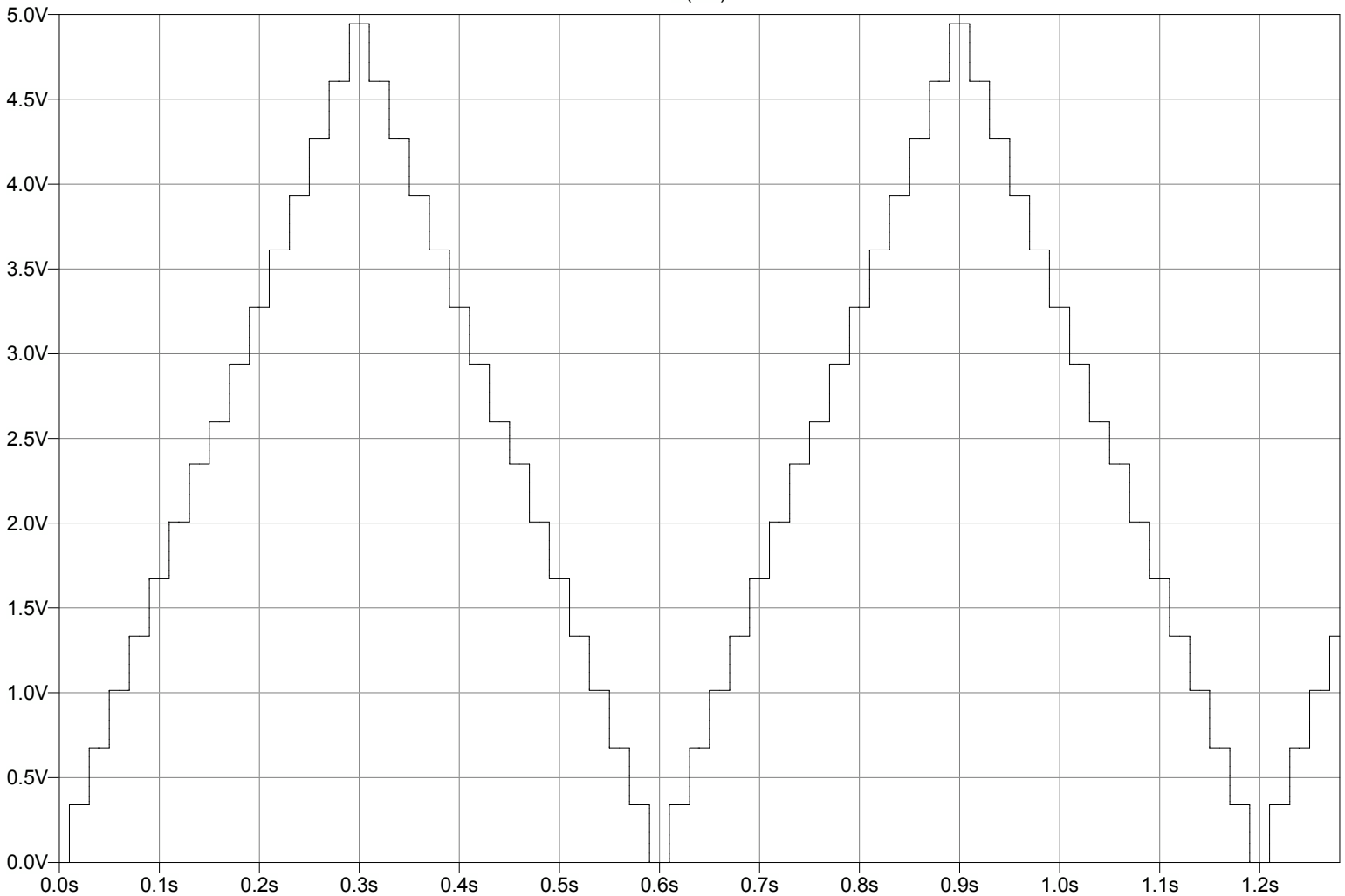
--- C:\Program Files\LTC\LTspiceIV\examples\MyCircuits\DAConverter.raw ---



`.include 74hc.lib .tran 1280ms .wave saw.wav 16 44.1k V(n001) V(Out)`

--- C:\Program Files\LTC\LTspiceIV\examples\MyCircuits\DAConverterUD.asc ---

V(out)

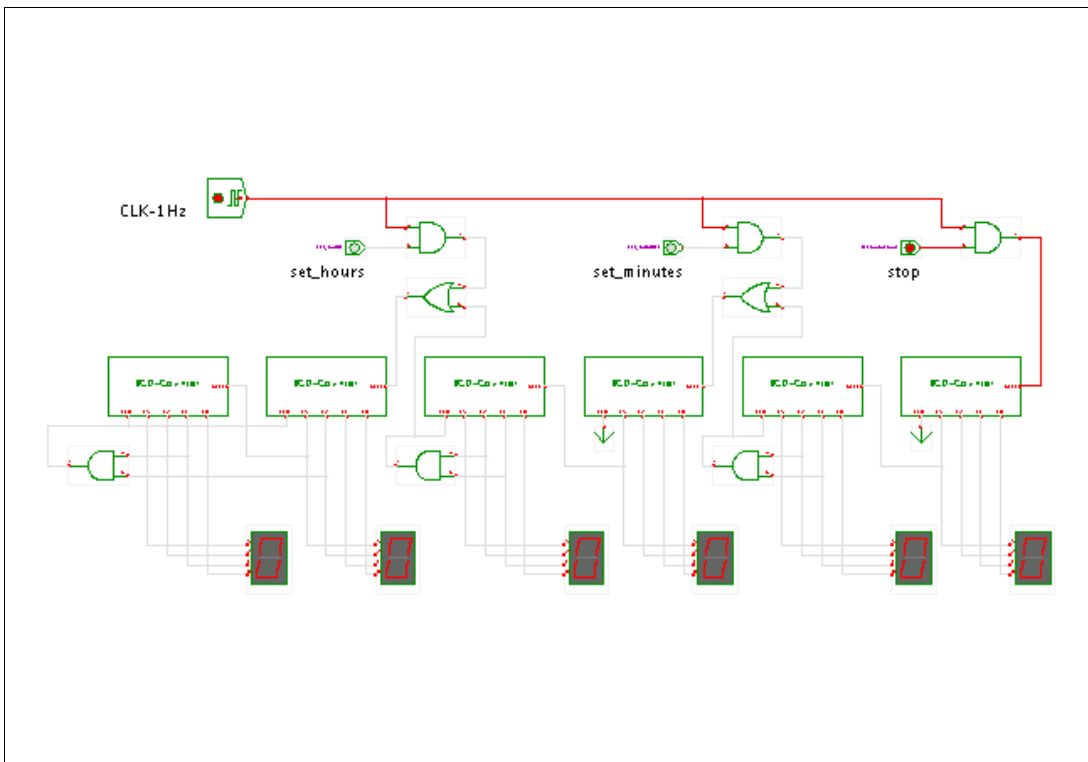


--- C:\Program Files\LTC\LTspiceIV\examples\MyCircuits\DAConverterUD.raw ---

Hades digital clock

[TAMS](#) / [Java](#) / [Hades](#) / **applets** (print version): [contents](#) | [previous](#) | [next](#)

Digital clock with BCD counters (settable)



Description

The same **digital clock** as shown in the previous applet, but with additional logic to set the time. The user interface consists of three switches, and is more or less typical for standalone digital clocks:

- click the *stop* switch to reset the seconds counter and pause the clock
- click the *set_minutes* switch to increment the minutes counter once per second
- click the *set_hours* switch to increment the hours counter once per second

Again, the circuit is built from a cascade of six asynchronous, falling-edge triggered BCD counters with a simple trick to reset the counters at the corresponding times.

The counter for the lower seconds digit simply runs continuously. The upper seconds digit counter, however, needs to be reset after 59 seconds to avoid counting to 99 seconds. This is achieved by connecting the asynchronous reset input of the counter block to its Z2 and Z1 outputs. Therefore, as soon as the counter reaches 60, it is immediately asynchronously reset and at the same time generates the carry to increment the minutes counter - which uses the same structure as the seconds counter.

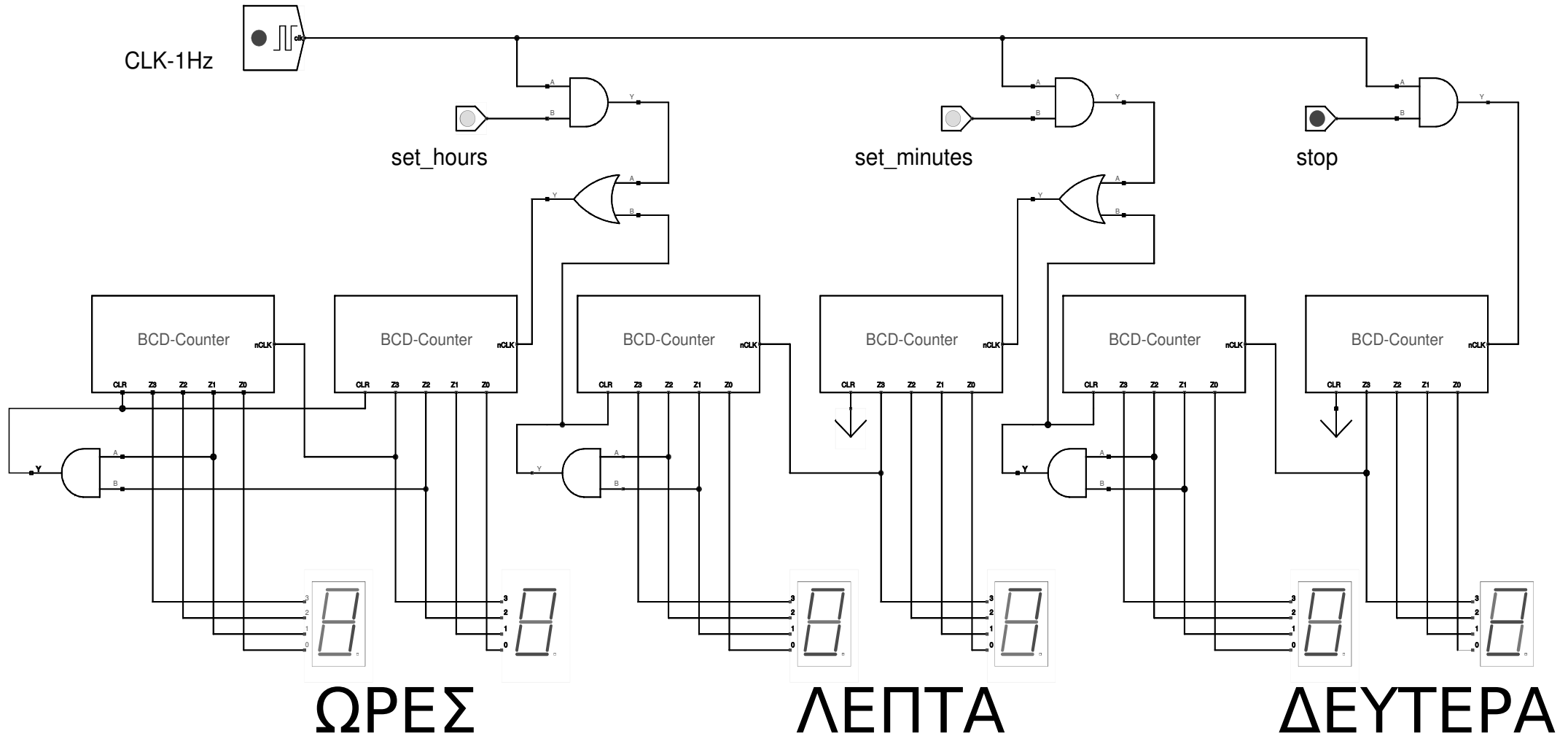
The hours counters use the same trick to immediately asynchronously reset the hours after the hours reach the value 24. Because the reset is asynchronous and very fast, the user has no chance to see the wrong values (like 00:00:60 or 24:00:00) before these are corrected to the right values.

[Run the applet](#) | [Run the editor](#) (via Webstart)

[Impressum](#) | 24.11.06

http://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/30-counters/80-digiclock/digiclock2_print.html

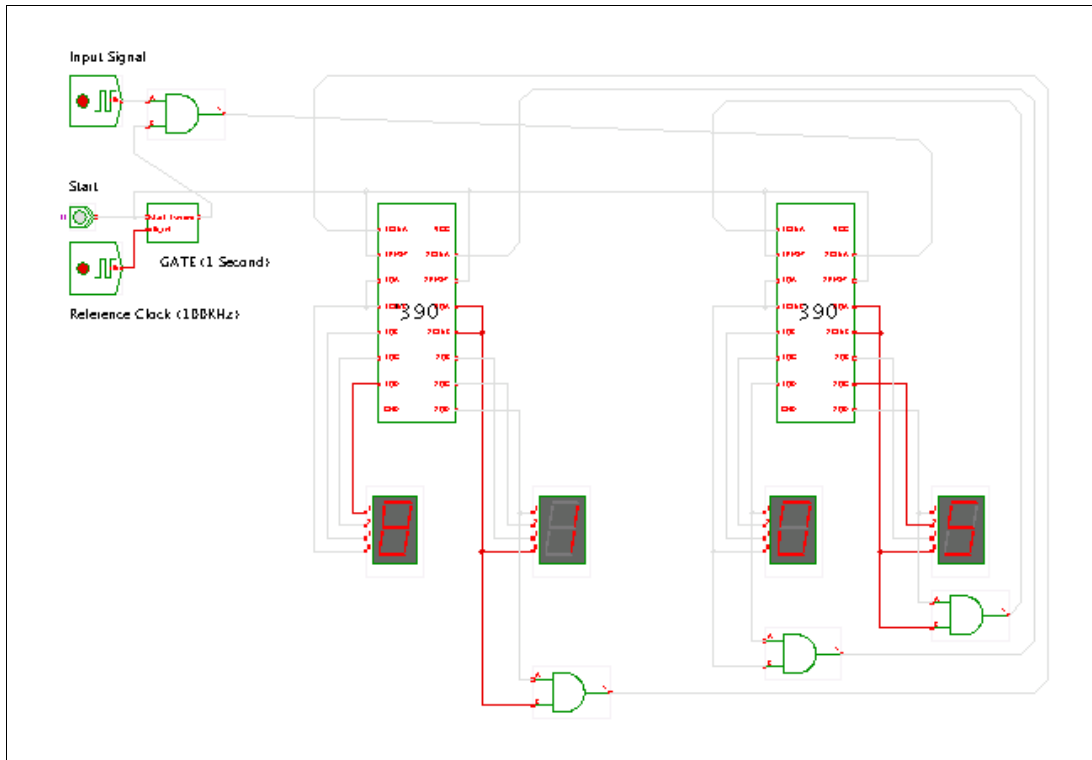
24 ΩΡΟ ΡΟΛΟΙ



Hades four-digit frequency counter

[TAMS](#) / [Java](#) / [Hades](#) / [applets](#) (print version): [contents](#) | [previous](#) | [next](#)

Four-digit frequency counter



Description

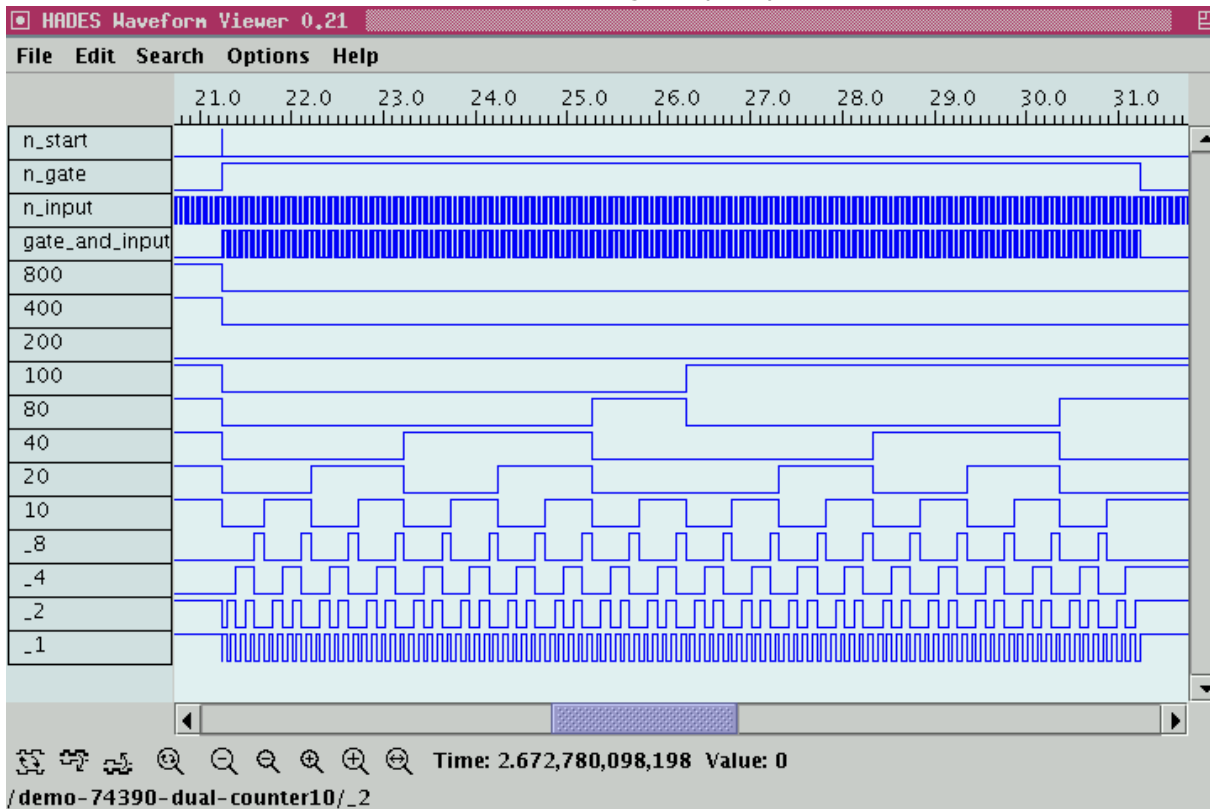
A **four-digit frequency counter** built from TTL-series 74390 integrated circuits.

We use a clock generator component to provide the input signal whose frequency should be measured. Use the property sheet of the upper-left clock generator (popup menu, edit component) to change the clock period. Click the 'start' switch (or press the 's' bindkey) to start a measurement.

As frequency is defined as 'pulses per second', it is sufficient to just count pulses during one second (or any other time interval) to measure the corresponding input frequency. To this end, a 'gate' component is used to limit the counting of input pulses to a time interval of known length. Obviously, when the time interval is exactly one second, no additional scaling is necessary, and the final counter value directly corresponds to the frequency of the input signal.

Pressing the 'start' button first resets the 4-digit decimal counter; it also initializes and starts the output pulse from the 'GATE' subdesign. This subdesign uses the reference clock of 100 KHz to create a single 1-second output pulse. (Changing the reference clock also changes the output pulse duration proportionally). The AND-gate masks the input signal to the 'GATE' pulse duration.

The following image shows typical example waveforms during a measurement:



See the next applet for the [internals of the GATE subdesign](#) which is also built from 74390 components. The idea is to create a highly accurate pulse duration in the range of seconds by counting thousands (millions) of short pulses, instead of trying to create a single pulse of the required length. While the latter is very difficult with electrical components, a cheap (temperature-stabilized) quartz-oscillator can be used to generate a very precise signal in the few-MHz range.

[Run the applet](#) | [Run the editor](#) (via Webstart)

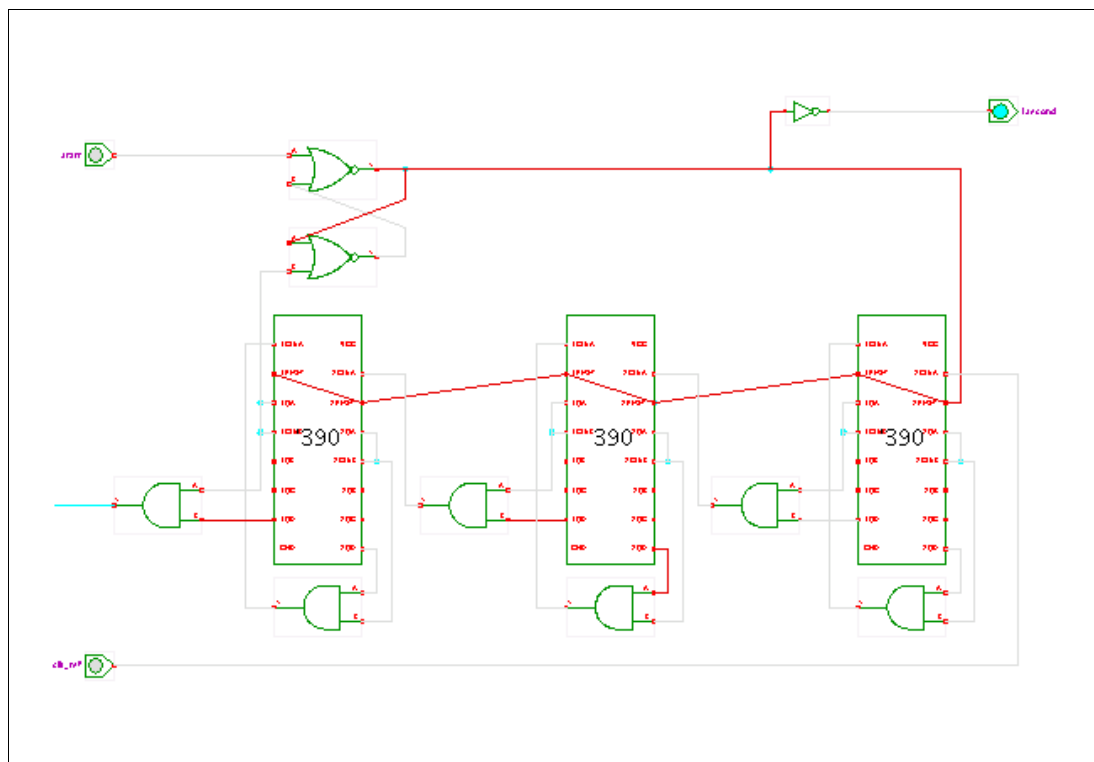
[Impressum](#) | 24.11.06

http://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/30-counters/70-ttl/frequency-counter_print.html

Hades counter-based pulse-generator

[TAMS](#) / [Java](#) / [Hades](#) / [applets](#) (print version): [contents](#) | [previous](#) | [next](#)

Counter-based pulse-generator



Description

This applet shows the **pulse-generator** sub-component required used in the [frequency counter](#) (see the previous applet).

The circuit consists of a NOR-gate RS-flipflop and a six-digit asynchronous decimal counter, built from 74390 integrated circuits. To avoid unnecessary repaints during simulation, no displays are used in this applet. Please see the description of the previous applets for details about the 74390 counter component and the decimal-carry logic via the AND-gates.

Pressing the 'start' switch will set the RS-flipflop and enable counting of the reference clock pulses. The output of the RS-flipflop is the 'gate' impulse (used to control the measurement-counter of the frequency counter). The NOR-gate RS-flipflop is reset when the counter value reaches the value 1000000 (one-million). Given the reference clock-generator period of $1.0E-5$ seconds, the output interval is exactly 1.00000 seconds.

Note that the pulse-generator only works as long as the 'start' switch is reset before the counter overflows. Please make sure to press the 'start' switch off again soon enough after starting the counter. In a 'real' pulse-generator, extra logic would have to be used to ensure this condition.

[Run the applet](#) | [Run the editor](#) (via Webstart)

[Impressum](#) | 24.11.06

http://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/30-counters/70-ttl/one-second_print.html