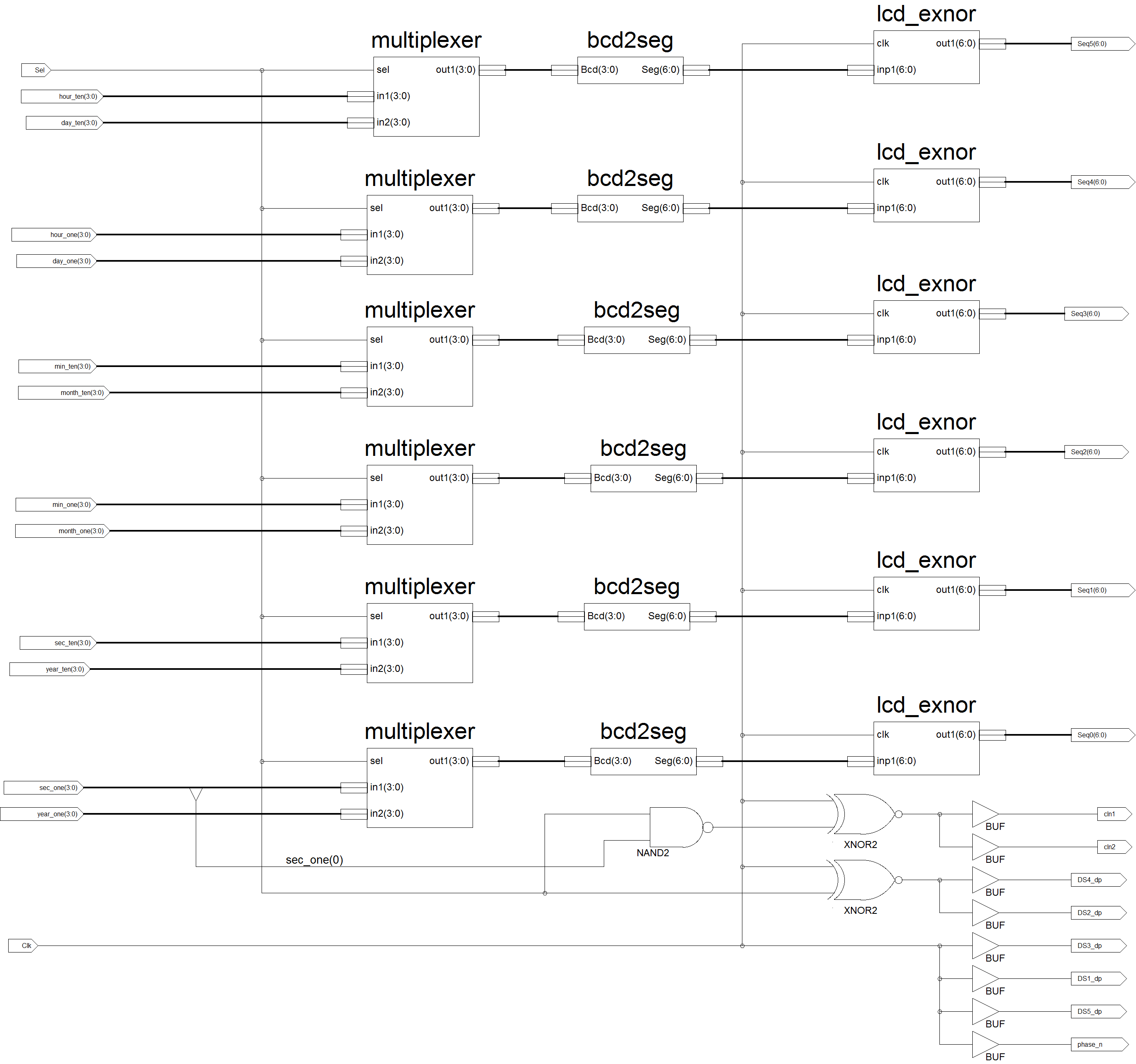
Display-Modul

Dieses Modul wird dazu benötigt, das 7-Segment-LCD-Display anzusteuern. Da das Display keine DC-Spannung verträgt, müssen alle Ansteuerungssignale getaktet werden.



# Inputs:

|  |  |
| --- | --- |
| clk : in STD\_LOGIC | Displayfrequenz (32Hz) |
| sel : in STD\_LOGIC | Select-Signal, 1 = Uhrzeit anzeigen  0 = Datum anzeigen |
| hour\_ten : in STD\_LOGIC\_VECTOR (3 downto 0); | Eingang Zehnerstelle Stunde |
| day\_ten : in STD\_LOGIC\_VECTOR (3 downto 0); | Eingang Zehnerstelle Tag |
| hour\_one : in STD\_LOGIC\_VECTOR (3 downto 0); | Eingang Einerstelle Stunde |
| day\_one : in STD\_LOGIC\_VECTOR (3 downto 0); | Eingang Einerstelle Tag |
| min\_ten : in STD\_LOGIC\_VECTOR (3 downto 0); | Eingang Zehnerstelle Minute |
| month\_ten : in STD\_LOGIC\_VECTOR (3 downto 0); | Eingang Zehnerstelle Monat |
| min\_one : in STD\_LOGIC\_VECTOR (3 downto 0); | Eingang Einerstelle Minute |
| month\_one : in STD\_LOGIC\_VECTOR (3 downto 0); | Eingang Einerstelle Monat |
| sec\_ten : in STD\_LOGIC\_VECTOR (3 downto 0); | Eingang Zehnerstelle Sekunde |
| year\_ten : in STD\_LOGIC\_VECTOR (3 downto 0); | Eingang Zehnerstelle Jahr |
| sec\_one : in STD\_LOGIC\_VECTOR (3 downto 0); | Eingang Einerstelle Sekunde |
| year\_one : in STD\_LOGIC\_VECTOR (3 downto 0); | Eingang Einerstelle Jahr |

# Outputs:

|  |  |
| --- | --- |
| Seg5 : out STD\_LOGIC\_VECTOR (6 downto 0); | Ausgang Segment 1 |
| Seg4 : out STD\_LOGIC\_VECTOR (6 downto 0); | Ausgang Segment 2 |
| Seg3 : out STD\_LOGIC\_VECTOR (6 downto 0); | Ausgang Segment 3 |
| Seg2 : out STD\_LOGIC\_VECTOR (6 downto 0); | Ausgang Segment 4 |
| Seg1 : out STD\_LOGIC\_VECTOR (6 downto 0); | Ausgang Segment 5 |
| Seg0 : out STD\_LOGIC\_VECTOR (6 downto 0); | Ausgang Segment 6 |
| cln1 : out STD\_LOGIC | Ausgang Doppelpunkt1 |
| cln2 : out STD\_LOGIC | Ausgang Doppelpunkt2 |
| DS1\_dp : out STD\_LOGIC | Ausgang Punkt zwischen Seg0 und Seg1 |
| DS2\_dp : out STD\_LOGIC | Ausgang Punkt zwischen Seg1 und Seg2 |
| DS3\_dp : out STD\_LOGIC | Ausgang Punkt zwischen Seg2 und Seg3 |
| DS4\_dp : out STD\_LOGIC | Ausgang Punkt zwischen Seg3 und Seg4 |
| DS5\_dp : out STD\_LOGIC | Ausgang Punkt zwischen Seg4 und Seg5 |
| phase\_n : out STD\_LOGIC | Gemeinsamer Anschluss LCD |

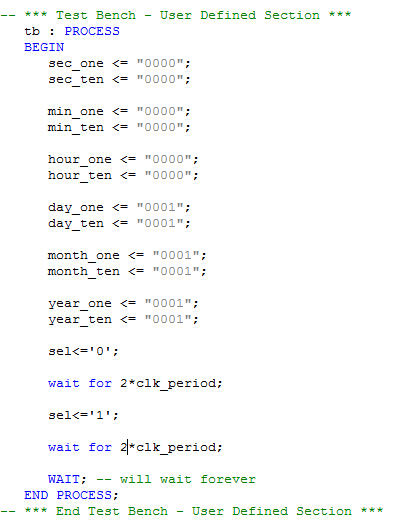
# Generic

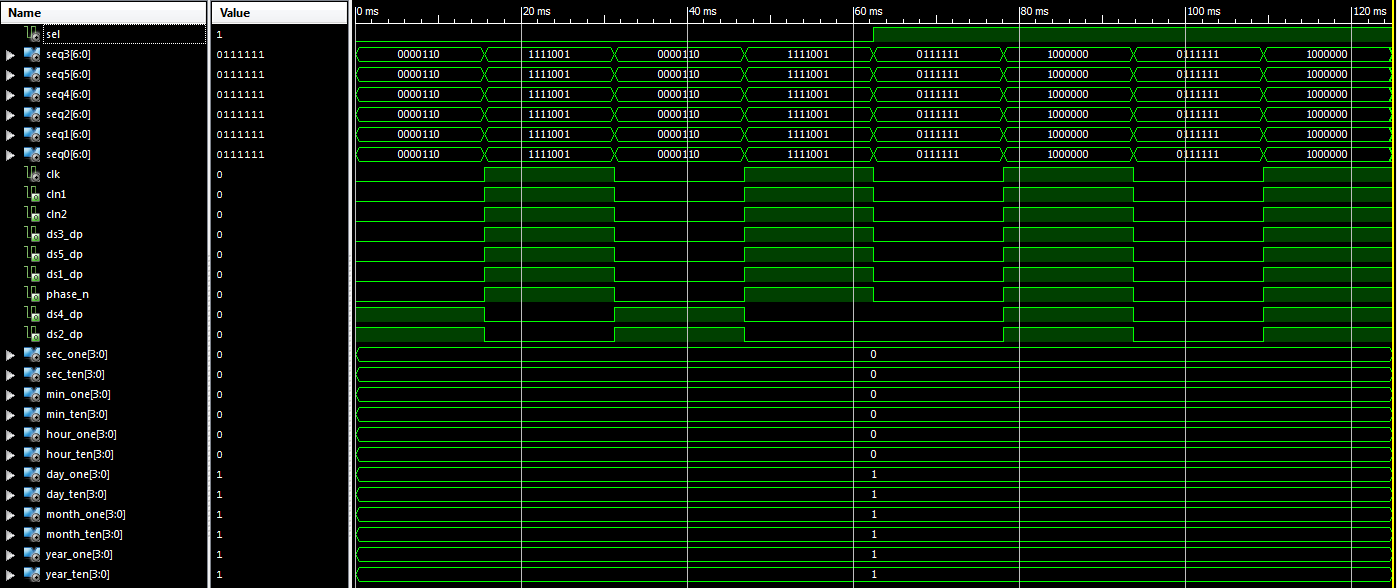
Keine Vorhanden.

# Beschreibung

Zuerst wird mit dem Multiplexer ausgewählt, welches Signal (Uhrzeit oder Datum) angezeigt werden soll. Danach wird aus dem 4Bit Breiten BDC-Code die Ansteuerungssignale für die einzelnen Segmente generiert. Danach werden die 7Bit Breiten Ansteuerungssignale für das LCD mit dem Takt EXNOR verknüpft und an das LCD ausgegeben. Der gemeinsame Anschluss des LCD (phase\_n) wird direkt an den Takt gelegt. Die Punkte 1, 3 und 5 werden auch an den Takt angelegt, da sie nie leuchten sollen. Die Doppelpunkte werden so verknüpft, dass sie im 0.5Hz Takt blinken nur wenn die Zeit dargestellt wird. Die beiden Punkte leuchten nur dann, wenn das Datum angezeigt wird.

# Testbench





Zuerst ist Sel = 0, das heisst das Datum 11.11.11 soll angezeigt werden. Während dieser Zeit ist ds4\_dp und ds2\_dp gegenphasig zu phase\_n. Damit leuchten die beiden Punkte beim Datum. Wenn der Takt ‚1‘ ist, sind bei den Segmenten nur gerade 2Bit ‚0‘, das heisst die beiden Segmente zur Darstellung der ‚1‘ leuchten. Wenn der Takt ‚0‘ ist sind die Bits gerade umgekehrt und zeigen auch wieder die ‚1‘ an.

Im zweiten Schritt wird Sel = 1, das heisst die Uhrzeit 00:00:00 soll angezeigt werden. Leider sieht man in dieser Darstellung nicht, wie die beiden Doppelpunkte blinken. Dies ist Fall, weil sie nur alle Sekunde invertiert werden und bei einer Simulationsdauer von 60ms ist dies nicht sichtbar. Wenn der Takt ‚1‘ ist, sind bei den Segmenten gerade 6Bit ‚0‘, das heisst die sechs Segmente zur Darstellung der ‚0‘ leuchten. Wenn der Takt ‚0‘ ist sind die Bits gerade umgekehrt und zeigen auch wieder die ‚0‘ an. Wie man sieht leuchten die beiden Punkte nicht mehr, das die gleichphasig wie der Takt (phase\_n) sind.