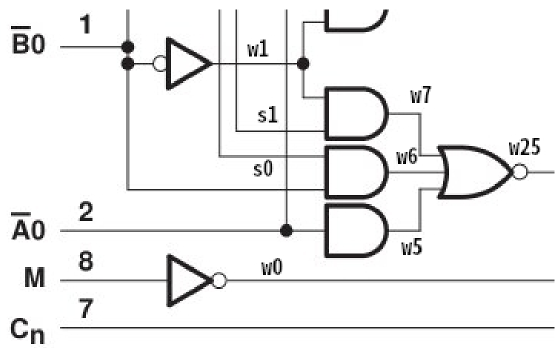
Lab: Specification of Texas Instrument sn74181 ALU

See the background information for the human readable specification of the TI sn74181. The ALU takes two 4-bit inputs, $\overline{\texttt{A0}}$, $\overline{\texttt{A1}}$, $\overline{\texttt{A2}}$, $\overline{\texttt{A3}}$, $\overline{\texttt{B0}}$, $\overline{\texttt{B1}}$, $\overline{\texttt{B2}}$, $\overline{\texttt{B3}}$ plus a carry bit `Cn`, and five selector bits, `S0`, `S1`, `S2`, `S3` and `M`. Depending on the selector bit values, various operations can be performed on the input bits with results appearing on output bits $\overline{\texttt{G}}$, `Cn+4`, $\overline{\texttt{P}}$, `A=B`, $\overline{\texttt{F0}}$. $\overline{\texttt{F1}}$, $\overline{\texttt{F2}}$, $\overline{\texttt{F3}}$. In the following the overbars will not be displayed for simplicity and clarity. In the following a functional specification and a wiring specification for the ALU will be developed and the two will be shown equivalent.

Exercise 1:
Write the specification for the wiring diagram. The following will be a helpful beginning. Consider the following piece of the wiring diagram taken from Page 4 of the TI spec sheet:



Labels have been given to the interconnect points. These labels represent new variables that have logic values as indicated in the wiring specification. For example, `w0` has value `~M` (use `~` for logical not), variable `w1` has value `~B0`, variable `w5` has value `A0` (we remove overbars), variable `w6` has value `s0 /\ B0`, `w7` has value `s1 /\ w1` and `w25` has value `~(w5 \/ w6 \/ w7)`. So, the description of the wiring can begin as a function like this:

```
f74181_netlist c a0 a1 a2 a3 b0 b1 b2 b3 m s0 s1 s2 s3 = [f0,f1,f2,f3,cout,p,g,a_b]
  where
    w0 = ~m
    w1 = ~b0
    w5 = a0
    w6 = s0 /\ b0
    w7 = s1 /\ w1
    w25 = ~(w5 \/ w6 \/ w7)
```

All ALU inputs are the arguments of the function `f74181_netlist`. The output will be a list of ALU outputs in an order that will match the output of a corresponding functional specification. In Cryptol `=` cannot be a part of a variable label so `a=b` is represent here as `a_b`. Similarly, `cout` is used to represent `Cn+4`. All overbars are removed and lower case letters for variables are used for simplicity and clarity.

Exercise 2:
The riddle of the Bitcoin and the Tigers. There are nine rooms, each with a sign on its door. A Bitcoin is in one room and one room only. Other rooms may be empty or have tigers. A prisoner has been tasked to open the door to the room containing the Bitcoin. If that happens the prisoner takes the Bitcoin and is released from prison. If a door containing a Tiger opened, the prisoner is eaten. If a door to an empty room is opened the prisoner is not eaten but remains in prison. The signs are:

ROOM 1: THE BITCOIN IS IN A ROOM WITH AN UNEVEN NUMBER
ROOM 2: THIS ROOM IS EMPTY
ROOM 3: EITHER SIGN 5 TELLS THE TRUTH, OR SIGN 7 IS LYING
ROOM 4: SIGN 1 IS LYING
ROOM 5: SIGN 2 IS LYING OR SIGN 4 TELLS THE TRUTH
ROOM 6: SIGN 3 IS LYING
ROOM 7: THE BITCOIN IS NOT IN ROOM 1
ROOM 8: THIS ROOM HOSTS A TIGER, AND ROOM 9 IS EMPTY
ROOM 9: THIS ROOM HOSTS A TIGER, AND SIGN 6 IS LYING

A sign on the door to a room containing the BITCOIN is True
A sign on the door to a room containing a TIGER is False
A sign on the door to an empty room could be True or False
If you know whether room 8 is empty, you uniquely know what room the Bitcoin is in

Write a function that enables finding the Bitcoin.

Note: If your function is called `FindBitcoin` then `:prove FindBitcoin` will confirm that the Bitcoin is found