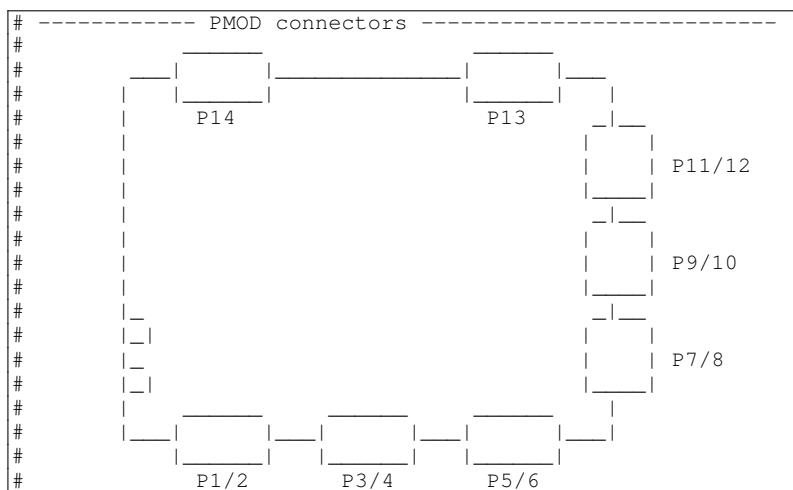


Génération de signal sonore

Utilisation d'un PMOD Audio

Vous connecterez le PMOD Audio sur le PMOD connecteur PMOD9/10 du BlackIceII.



Dans le fichier « blackice-ii.pcf » définissant les Pinout du FPGA vers les composants/connecteurs de la carte :

```
# -----| P10 |-----  
#  
#-----| 3V3 | GND | P93  | P92  | P91  | P90  |-----^  
#-----| 3V3 | GND | P103 | P102 | P101 | P100 | <  
#-----  
  
set_io --warn-no-port P100 18  
set_io --warn-no-port P101 17  
set_io --warn-no-port P102 12  
set_io --warn-no-port P103 11
```

La broche utilisée pour le « speaker » est la P100 sur le schéma, soit la 18 pour le FPGA.

- 1 –** a. Ajoutez la définition du signal « speaker » dans le fichier de contraintes de broches « .pcf » ;
b. Vous récupérerez le code à https://git.p-fb.net/PeFClic/fpga_pmodaudio
c. Pour synthétiser le circuit, vous utiliserez le « Makefile » de la manière suivante :

```
□ — xterm —  
$ make VERILOG_SRC=pmod_audio.v TOPLEVEL=pmod_audio
```

et pour programmer :

```
$ make VERILOG_SRC=pmod_audio.v TOPLEVEL=pmod_audio prog
```

Le fichier Makefile est configuré par des variables passées en paramètre.

- d. Comment est généré le signal à destination du speaker :

À quoi sert le registre « counter » ? Est-il de bonne taille ?

Qu'est-ce que le « *duty cycle* » et pourquoi doit-on diviser par 2 ?

```

// This module generates a PWM signal with a variable frequency.
// The frequency is determined by the FREQUENCY parameter.
// The PWM signal is output on the PWM_OUT pin.

module pmod_audio #(
    parameter FREQUENCY = 440 // 440 Hz is the frequency of the note A4
) (
    input wire clk,
    input wire greset,
    output wire speaker
);
    // The counter will increment at each clock cycle.
    reg [31:0] counter;

    // The period is the number of clock cycles in one period of the PWM signal.
    // It is calculated by dividing the clock frequency by the desired frequency.
    // For example, if the clock frequency is 100 MHz and the desired frequency is
    440 Hz,
    // the period will be 100,000,000 / 440 = 227272.
    localparam PERIOD = 100_000_000 / FREQUENCY;

    // The duty cycle is the percentage of time that the PWM signal is high.
    // In this case, we want a 50% duty cycle, so the duty cycle is half of the per
iod.
    localparam DUTY_CYCLE = PERIOD / 2;

    // The PWM signal is generated by comparing the counter to the duty cycle.
    // If the counter is less than the duty cycle, the PWM signal is high.
    // Otherwise, the PWM signal is low.
    assign speaker = (counter < DUTY_CYCLE);

    // The counter is incremented at each clock cycle.
    // If the counter reaches the period, it is reset to 0.
    always @ (posedge clk) begin
        if (!greset) begin
            counter <= 0;
        end else begin
            if (counter == PERIOD) begin
                counter <= 0;
            end else begin
                counter <= counter + 1;
            end
        end
    end
endmodule

```

- 2–** Vous modifierez le circuit pour alterner entre une fréquence de 440 et de 100 hertz toutes les demi-secondes.

Comment allez vous procéder en réutilisant le circuit précédent ?

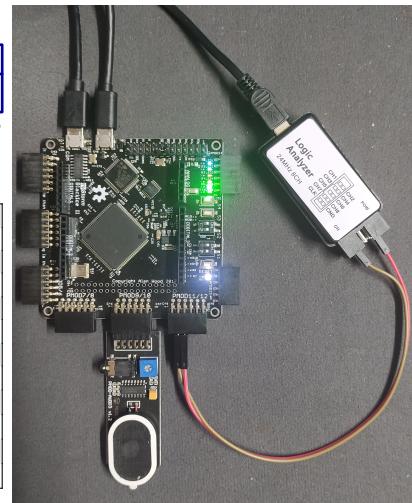
■ ■ ■ Utilisation d'un analyseur logique

Vous installerez les paquets sigrok et pulseview:

— xterm

```
$ sudo apt install sigrok pulseview
```

Vous connecterez l'analyseur logique vers les broches GND et P113 du PMOD11/12.



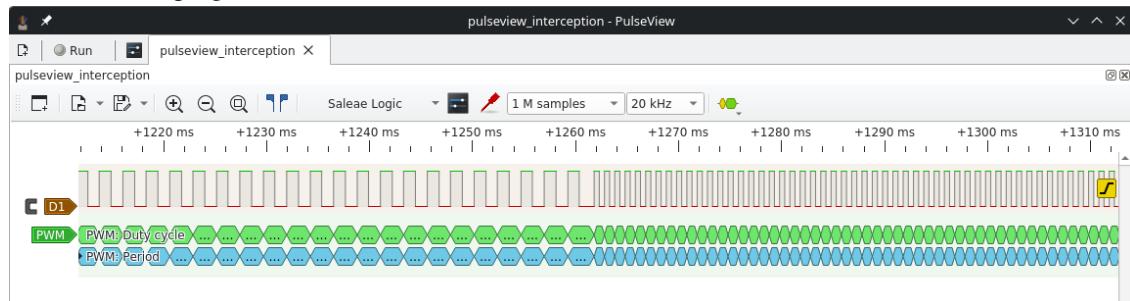
Vous ajouterez la broche «analyseur» dans le fichier «.pcf» pour la broche P113.

Vous connecterez sur l'analyseur logique :

- ▷ le « GND » du connecteur du FPGA au « CND » de l'analyseur (C ou G, quelle importance...);
 - ▷ la broche P113 du FPGA vers la broche CH2 de l'analyseur;

- 3 – a.** Comment allez vous modifier le circuit pour diriger le signal allant vers le speaker vers la broche allant sur l'analyseur logique ?

En exécutant le programme « pulseview » :



- ▷ Vous activerez uniquement la broche « D1 », le logiciel commençant à numérotter les broches à zero.
 - ▷ Vous ajouterez un décodage en PWM sur cette broche qui vous affichera :
 - ◊ le duty cycle ;
 - ◊ la période ;
 - b. À quelle fréquence l'analyseur logique doit échantillonner la sortie du FPGA ?
 - c. Avec ces informations, pouvez-vous retrouver les fréquences générées par votre circuit ?