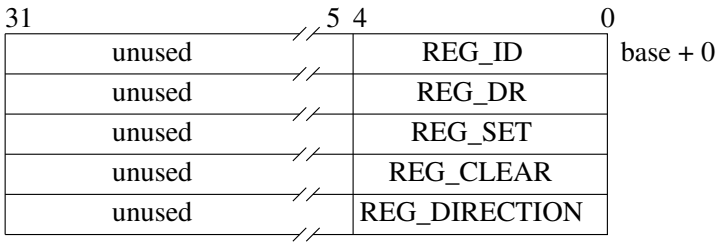
Électronique embarquée

Les questions en caractères gras valent 2 points. Tous documents autorisés.

Deux contrôleurs de 4 bits associés aux LEDs et 1 bit associé à une broche sur le port E1 sont configurés dans le FPGA au moyen du bitstream disponible dans l'archive à `/home/jmfriedt/exam2021_M2_ElecEmbarquee.tar.gz`. Dans un premier temps, on ne s'intéressera qu'au premier contrôleur présent à l'adresse de base 0x43C00000.

L'organisation des registres est la même que lors de l'examen de Systèmes Embarqués, à savoir des registres codés sur 32 bits contigus avec un identifiant dans le premier registre `REG_ID`, suivi de la valeur attribuée aux sorties `REG_DR`, puis le registre permettant d'imposer certains bits à 1 par un masque dans `REG_SET`, suivi du même effet mais pour imposer à 0 dans `REG_CLEAR` et finalement la définition de la direction de chaque bit dans `REG_DIRECTION`. Le premier contrôleur est relié aux LEDs 0 à 3 pour les bits de poids faibles et E1_3 pour le bit cinquième bit.



1. Peut-on accéder directement à une adresse physique dans la mémoire d'un processeur ARM tel que celui qui équipe le processeur Zynq de la Redpitaya. Proposer la syntaxe correspondante en C pour justifier votre réponse. **Compléter la fonction d'initialisation du squelette de module en conséquent.**
2. Où se trouve le compilateur dans l'arborescence de compilation croisée Buildroot ?
3. Où se trouvent les sources du noyau Linux dans l'arborescence de Buildroot ?
4. **Proposer un module noyau qui permette d'accéder aux registres de l'IP GPIO configurant le FPGA. Afficher dans les logs du noyau l'identifiant unique de l'IP (REG_ID).** On rappelle que pour charger la configuration du FPGA, on placera le bitstream d'extension `.bit.bin` dans `/lib/firmware` et on informe FPGA Manager de son utilisation en écrivant le nom du bitstream dans `/sys/class/fpga_manager/fpga0/firmware`
5. Quelles sont les catégories d'interfaces de communication dans `/dev` et qu'est-ce qui les caractérise ? Quelle est cette caractéristique pour le point de communication créé par le squelette de module que nous proposons ?
6. Quelle est la direction par défaut des broches du GPIO ?
7. **Proposer deux fonctions de communication via l'interface dans `/dev` pour définir l'état des GPIOs par une écriture dans `REG_DR` et pour en lire le contenu.**
8. Démontrer le bon fonctionnement de ces communications par un programme en C depuis l'espace utilisateur qui fasse clignoter les LEDs, alternativement allumant les LEDs d'indice pair et éteignant les LEDs d'indice impair et échangeant leur état chaque seconde.
9. **Quel mécanisme d'un module noyau communiquant par `dev` permet de configurer la direction des broches ? Proposer une implémentation dans votre module.**
10. Compléter le programme C en espace utilisateur en conséquence afin de passer le cinquième bit en entrée.
11. Lire depuis le programme en C exécuté en espace utilisateur l'état de cette broche lorsqu'un fil la relie une à la masse. Idem en reliant à 3,3 V.

Afin de généraliser ce que nous venons de voir, un squelette de *platform device* est fourni. Sachant que le FPGA intègre une deuxième IP GPIO – la première que nous venons de contrôler par `/dev` et une seconde à l'adresse 0x43C00020 reliée aux LEDs 4 à 7 et à E1_4 pour le cinquième bit – compléter le squelette avec les fonctions permettant d'accéder aux ressources matérielles de chaque bloc GPIO en passant les ressources comme argument au pilote lors de l'instantiation du périphérique. Pour ce faire :

12. **On commencera par récupérer l'indice (`REG_ID`) dans la fonction appelée au chargement du pilote (quelle est-elle)** ...
13. ... **pour compléter avec une interface de communication en écriture depuis l'espace utilisateur pour définir la direction des divers bits du GPIO** ...
14. ... pour compléter avec une interface d'écriture pour définir l'état des LEDs ...
15. ... **et une interface de lecture pour consulter le statut du bit en entrée. Démontrer le bon fonctionnement sur les deux IP GPIO.**