



SignalTap II avec conceptions VHDL

Pour Quartus II 13.0

1 Introduction

Ce tutoriel explique comment utiliser la fonctionnalité SignalTap II du logiciel Quartus II d'Altera. L'analyseur logique[®] intégré SignalTap II est un outil de débogage système qui capture et affiche les signaux des circuits conçus pour être implémentés dans les FPGA d'Altera.

Contenu:

- Exemple de circuit
- Activation de la fonction TalkBalk de Quartus II
- Utilisation de l'analyseur logique SignalTap II
- Sonder la conception à l'aide de SignalTap
- Options de déclenchement avancées
- Modes d'acquisition de la profondeur d'échantillon et du tampon

2 Contexte

Le logiciel Quartus® II comprend un outil de débogage au niveau du système appelé SignalTap II qui peut être utilisé pour capturer et afficher des signaux en temps réel dans n'importe quelle conception FPGA.

Au cours de ce tutoriel, le lecteur découvrira :

- Sondage des signaux à l'aide du logiciel SignalTap
- Configuration de déclencheurs pour spécifier quand les données doivent être capturées

Ce tutoriel s'adresse aux lecteurs souhaitant sonder des signaux dans des circuits définis avec le langage de description matérielle VHDL. Un tutoriel équivalent est disponible pour les lecteurs préférant le langage Verilog.

Le lecteur doit disposer d'un ordinateur équipé du logiciel Quartus II. Les exemples détaillés de ce tutoriel ont été obtenus avec Quartus II version 13.0, mais d'autres versions du logiciel peuvent également être utilisées.

Note:

Veuillez noter qu'il n'y a pas de LED rouges sur les cartes DE0 ou DE0-Nano. Toutes les procédures utilisant des LED rouges dans ce tutoriel doivent être effectuées sur les cartes DE0 ou DE0-Nano en utilisant des LED vertes. Si vous suivez ce tutoriel sur une carte DE0, remplacez LEDR par LEDG dans les modules Verilog ci-dessous. Si vous suivez ce tutoriel sur une carte DE0-Nano, remplacez LEDR par LED ci-dessous.

De plus, le DE0-Nano est limité à 4 commutateurs. Si vous suivez ce tutoriel sur un DE0-Nano, remplacez toutes les occurrences de (7 DOWNTO 0) par (3 DOWNTO 0) ci-dessous.

3 exemples de circuits

À titre d'exemple, nous utiliserons le circuit de commutation implémenté en VHDL de la figure 1. Ce circuit connecte simplement les 8 premiers commutateurs de la carte série DE aux 8 premières LED rouges de la carte. Il le fait au front positif de l'horloge (CLOCK_50) en chargeant les valeurs des commutateurs dans un registre dont la sortie est directement connectée aux LED rouges.

Implémentez ce circuit comme suit :

- Créer un projet switches.
- Inclure un fichier switches.vhd, qui correspond à la figure 1, dans le projet.
- Sélectionnez le périphérique associé à la carte de la série DE. Liste des périphériques de la série DE.

Les tableaux peuvent être trouvés dans le tableau 1.

Importez le fichier qsf correspondant. Par exemple, pour une carte DE2, ce fichier s'appelle DE2_pin_assignments.qsf et peut être importé en cliquant sur Affectations > Importer les affectations. Pour plus de commodité, ce fichier est fourni dans le

Sous-répertoire design_files du dossier tutorials, inclus sur le CD-ROM accompagnant la carte DE-series et également disponible sur les pages web d'Altera dédiées à la série DE. Les noms de nœuds utilisés dans l'exemple de circuit correspondent à ceux de ce fichier.

Sur une carte DE2-70, modifiez le mode de fonctionnement de la broche nCEO en E/S standard en allant dans Affectations > Périphérique > Options de périphérique et de broche > Broches à double usage, puis double-cliquez sur le champ Valeur de la broche nCEO et définissez-le sur Utiliser comme E/S standard. La broche nCEO peut être réservée comme broche de programmation dédiée ou comme broche d'E/S standard. Pour les besoins de ce tutoriel, nous l'utiliserons comme broche d'E/S standard.

- Compiler la conception.

```

BIBLIOTHÈQUE ieee ;
UTILISER ieee.std_logic_1164.all ;
ENTITÉ commutateurs IS
    PORT ( CLOCK_50 : IN                      STD_LOGIC;
           Sud-Ouest : DANS STD_LOGIC_VECTOR(7 JUSQU'À 0);
           LEDR      : OUT STD_LOGIC_VECTOR(7 DOWNT0 0)); -- LED rouges
Interrupteurs END ;
ARCHITECTURE Comportement des commutateurs IS
COMMENCER
    PROCESSUS (CLOCK_50)
        COMMENCER
            SI(FRONT_ASCENDANT(HORLOGE_50)) ALORS
                LEDR <= SW;
            FIN SI;
        FIN DU PROCESSUS;
    FIN Comportement ;

```

Figure 1. Le circuit de commutation implémenté dans le code VHDL

Conseil	Nom de l'appareil
DE0	Cyclone III EP3C16F484C6
DE0-Nano	Cyclone IVE EP4CE22F17C6
DE1	Cyclone II EP2C20F484C7
DE2	Cyclone II EP2C35F672C6
DE2-70	Cyclone II EP2C70F896C6
DE2-115	Cyclone IVE EP4CE115F29C7

Tableau 1. Noms des périphériques FPGA de la série DE

4 Activation de la fonction TalkBack du Quartus II

Lors de l'utilisation de Quartus II Web Edition, la fonctionnalité TalkBack doit être activée pour accéder au logiciel SignalTap II. L'activation de TalkBack permet à Quartus II et aux autres programmes Altera d'envoyer à Altera une quantité limitée de données concernant leur utilisation. Ces données servent principalement à mieux comprendre les interactions des utilisateurs avec le logiciel et n'incluent aucun fichier de conception. La liste complète des données envoyées ou non est disponible dans le contrat de licence TalkBack.

Pour activer la fonctionnalité TalkBack dans Quartus II, sélectionnez Outils > Options. Dans la fenêtre Options, sélectionnez « Connectivité Internet » dans le menu, puis cliquez sur « Options TalkBack... » pour ouvrir la fenêtre illustrée à la figure 2. Si vous acceptez le contrat de licence TalkBack, cochez la case « Activer l'envoi de données TalkBack à Altera » et cliquez sur « OK ».

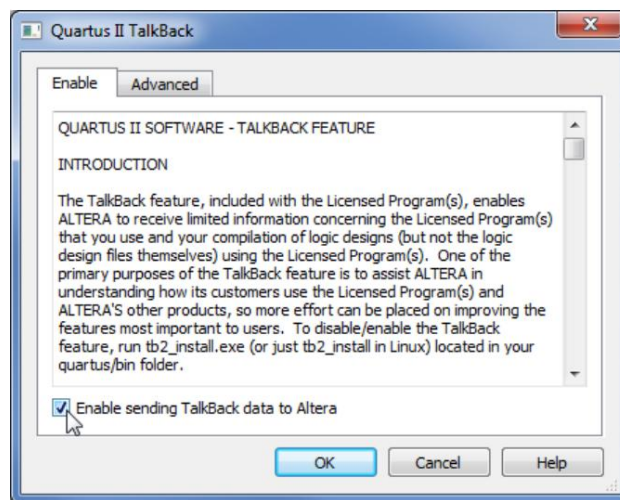


Figure 2. Activation de la fonction TalkBack.

5 Utilisation du logiciel SignalTap II

Dans la première partie du tutoriel, nous allons configurer l'analyseur logique SignalTap pour sonder les valeurs des 8 commutateurs LED. Nous allons également configurer le circuit pour qu'il se déclenche lorsque le premier commutateur (LED[0]) est à l'état haut.

1. Ouvrez la fenêtre SignalTap II en sélectionnant Fichier > Nouveau, ce qui donne la fenêtre illustrée dans la Figure 3. Choisissez Fichier d'analyseur logique SignalTap II et cliquez sur OK.

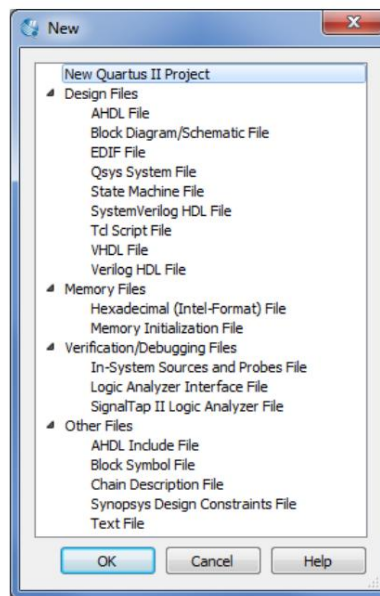


Figure 3. Besoin de préparer un nouveau fichier.

2. La fenêtre SignalTap II, avec l'onglet Configuration sélectionné, est illustrée à la figure 4. Enregistrez le fichier sous le nom switches.stp. Dans la boîte de dialogue qui s'affiche (figure 5), cliquez sur OK. Dans la boîte de dialogue « Voulez-vous activer le fichier SignalTap II 'switches.stp' pour le projet actuel ? », cliquez sur Oui (figure 6). Le fichier switches.stp est désormais le fichier SignalTap associé au projet.

Remarque : Si vous souhaitez désactiver ce fichier du projet ou désactiver SignalTap, accédez à Affectations > Paramètres. Dans la liste des catégories, sélectionnez l'analyseur logique SignalTap II pour afficher la fenêtre illustrée à la figure 7. Pour désactiver l'analyseur, décochez la case « Activer l'analyseur logique SignalTap II ». Il est également possible d'avoir plusieurs fichiers SignalTap pour un projet donné, mais un seul peut être activé à la fois. Avoir plusieurs fichiers SignalTap peut être utile si le projet est très volumineux et que différentes sections doivent être analysées. Pour créer un fichier SignalTap pour un projet, répétez simplement les étapes 1 et 2 et attribuez un nom différent au nouveau fichier. Pour modifier le fichier SignalTap associé au projet, dans la zone Nom du fichier SignalTap II, recherchez le fichier souhaité, cliquez sur Ouvrir, puis sur OK. Pour ce tutoriel, nous souhaitons laisser SignalTap activé et nommer le fichier SignalTap II switches.stp. Vérifiez que c'est bien le cas et cliquez sur OK pour quitter la fenêtre des paramètres.

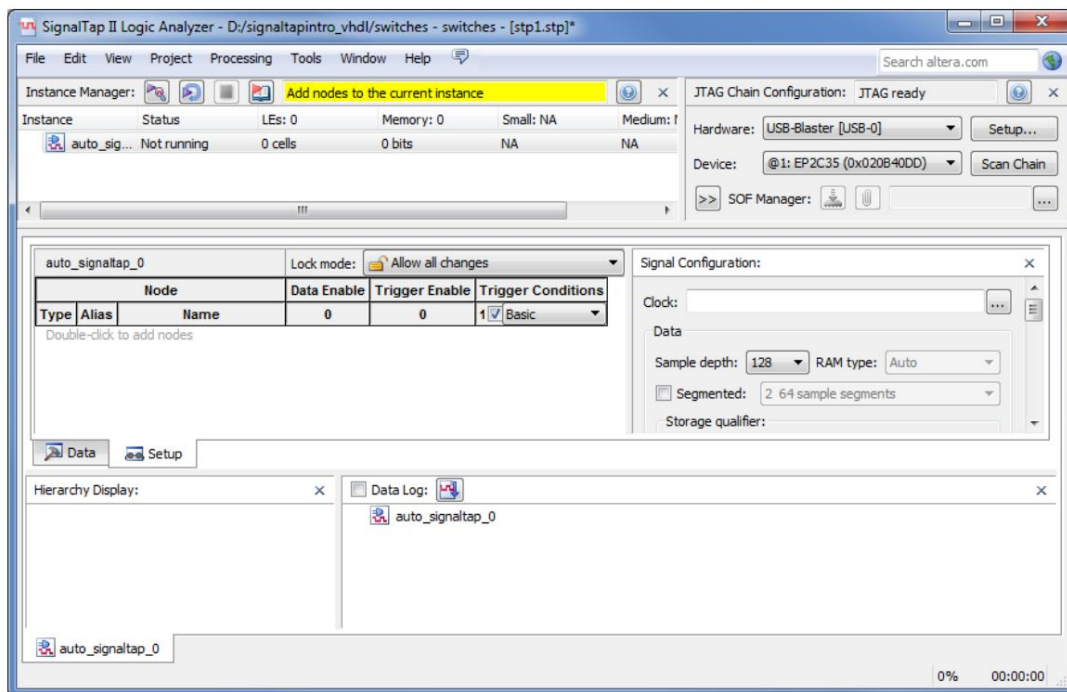


Figure 4. La fenêtre SignalTap II.

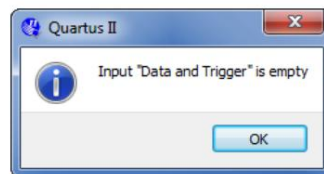


Figure 5. Cliquez sur OK dans cette boîte de dialogue.



Figure 6. Cliquez sur Oui dans cette boîte de dialogue.

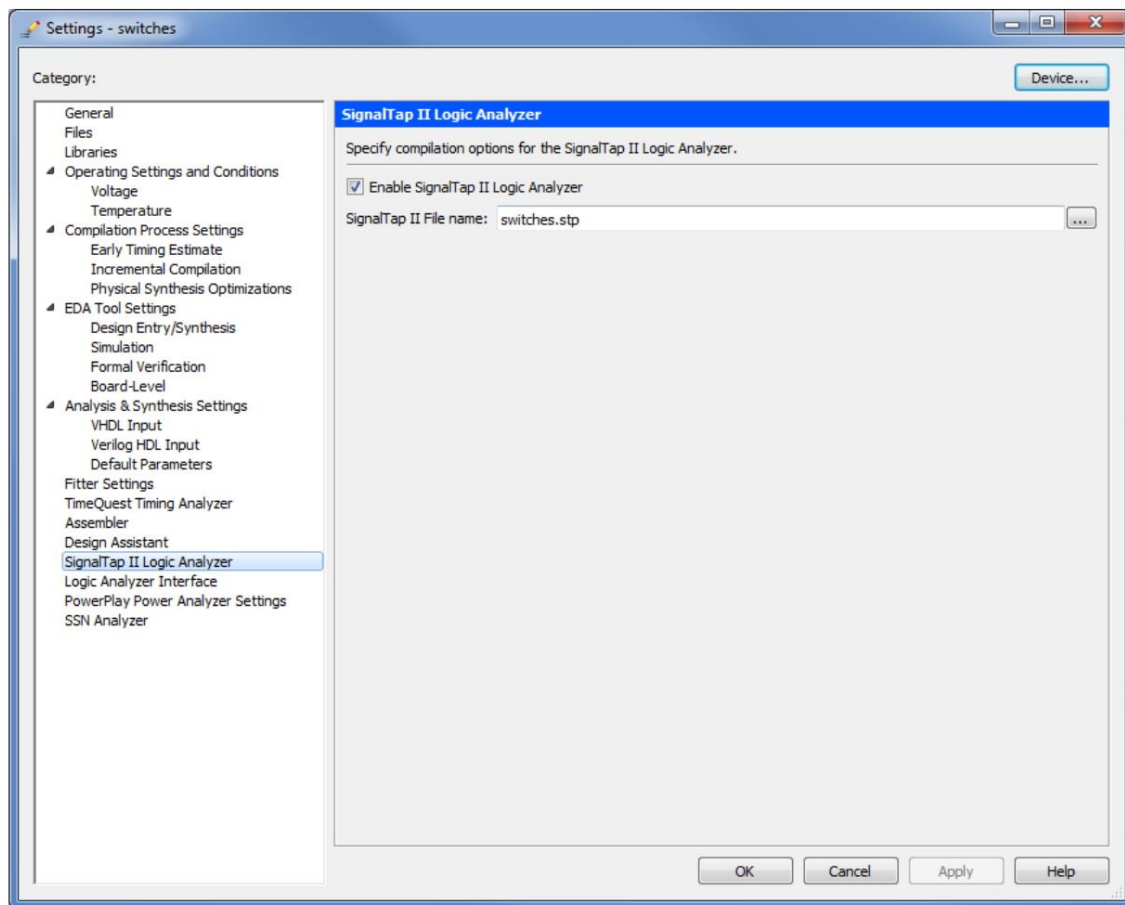



Figure 7. La fenêtre Paramètres de SignalTap II.

3. Nous devons maintenant ajouter les nœuds du projet à sonder. Dans l'onglet Configuration de la fenêtre SignalTap II, double-cliquez sur la zone « Double-cliquez pour ajouter des nœuds » pour ouvrir la fenêtre de recherche de nœuds, comme illustré à la figure 8. Cliquez sur  pour afficher plus d'options de recherche. Dans le champ Filtre, sélectionnez SignalTap II : pré-synthèse et, dans le champ Rechercher dans, sélectionnez [switches]. Cliquez sur Liste. Tous les nœuds du projet pouvant être sondés s'affichent. Sélectionnez SW[0] à SW[7], puis cliquez sur le bouton > pour ajouter les commutateurs à sonder. Cliquez ensuite sur OK.

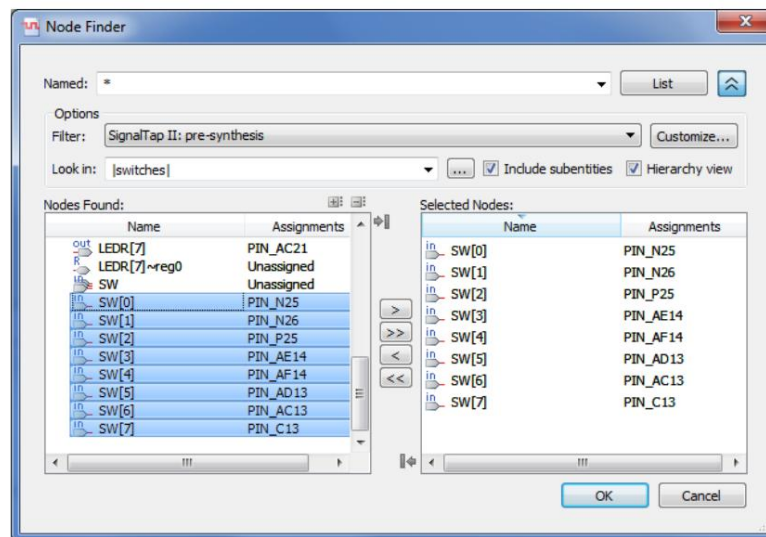


Figure 8. Ajout de nœuds dans la fenêtre Node Finder sur une carte de la série DE.

4. Avant que l'analyseur SignalTap puisse fonctionner, nous devons spécifier quelle horloge va exécuter le module SignalTap qui sera instancié dans notre conception. Pour ce faire, dans la zone Horloge du panneau Configuration du signal de l' Dans la fenêtre SignalTap, ..., qui ouvrira à nouveau la fenêtre de recherche de nœuds. Sélectionnez Liste pour afficher tous les nœuds. Cliquez sur les nœuds qui peuvent être ajoutés comme horloge, puis double-cliquez sur CLOCK_50, ce qui donne l'image affichée dans Figure 9. Cliquez sur OK.

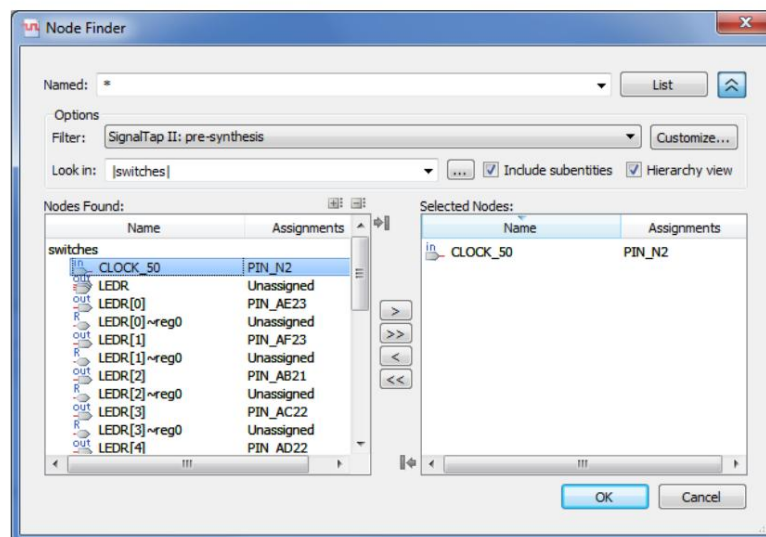


Figure 9. Définition de CLOCK_50 comme horloge pour l'instance SignalTap sur une carte de la série DE.

5. Avec l'onglet Configuration de la fenêtre SignalTap sélectionné, cochez la case dans la colonne Conditions de déclenchement. Dans le menu déroulant en haut de cette colonne, sélectionnez « Basique ». Faites un clic droit sur la cellule « Conditions de déclenchement ».

correspondant au nœud SW[0] et sélectionnez « Élevé ». L'exécution de l'analyseur logique sera alors déclenchée lorsque le premier commutateur de la carte série DE sera réglé sur « Élevé », comme illustré à la figure 10. Vous pouvez faire un clic droit sur la cellule « Conditions de déclenchement » de n'importe quel nœud sondé et sélectionner la condition de déclenchement parmi plusieurs options. La condition de déclenchement sera vraie lorsque l'opération « ET » logique de toutes ces conditions sera satisfaite. Pour l'instant, conservez la condition de déclenchement SW[0] sur « Élevé » et les autres sur leur valeur par défaut, « Indifférent ».

auto_singaltap_0			Lock mode: Allow all changes		
Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	8	8	1 Basic
		SW[0]			
		SW[1]			
		SW[2]			
		SW[3]			
		SW[4]			
		SW[5]			
		SW[6]			
		SW[7]			

Figure 10. Définition des conditions de déclenchement.

6. Pour que SignalTap II fonctionne, nous devons configurer correctement le matériel. Tout d'abord, assurez-vous que la carte série DE est branchée et sous tension. Dans la section Matériel de la fenêtre SignalTap II, située en haut à droite, cliquez sur Configuration pour afficher la fenêtre de la figure 11. Double-cliquez sur USB-Blaster dans le menu « Matériels disponibles », puis cliquez sur Fermer.

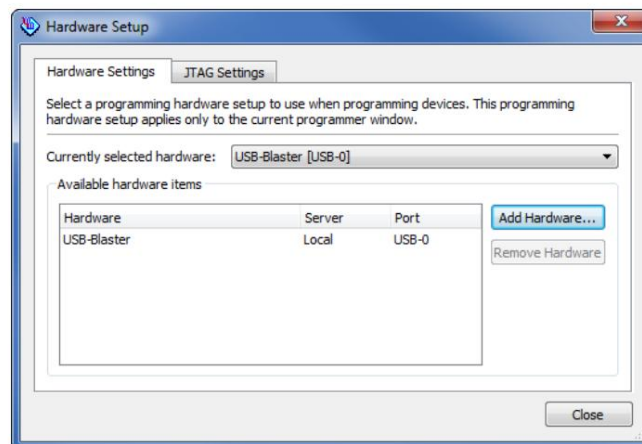



Figure 11. Configuration du matériel.

7. La dernière étape de l'instanciation de SignalTap dans votre conception consiste à compiler celle-ci. Dans la fenêtre principale de Quartus II, sélectionnez Traitement > Démarrer la compilation et indiquez que vous souhaitez enregistrer les modifications du fichier en cliquant sur Oui. Après la compilation, accédez à Outils > Programmeur et chargez le projet sur la carte de la série DE.

6 Sondage de la conception à l'aide de SignalTap II

Maintenant que le projet avec SignalTap II instancié a été chargé sur la carte de la série DE, nous pouvons sonder les nœuds comme nous le ferions avec un analyseur logique externe.

1. Sur la carte série DE, commencez par régler tous les commutateurs (0 à 7) sur « bas ». Nous allons essayer de tester leurs valeurs. commute une fois que le commutateur 0 devient haut.

2. Dans la fenêtre SignalTap, sélectionnez Traitement > Exécuter l'analyse ou cliquez sur l'icône  Cliquez ensuite sur l'onglet Données de la fenêtre SignalTap II. Un écran similaire à la figure 12 devrait s'afficher. Notez que la colonne d'état du panneau Gestionnaire d'instances SignalTap II indique « En attente de déclenchement ». Ceci est dû au fait que la condition de déclenchement (commutateur 0 à l'état haut) n'est pas encore remplie. (Ceci est valable si le commutateur 0 est à l'état bas, comme indiqué à l'étape précédente. Si ce n'est pas le cas, réglez-le sur bas, puis cliquez à nouveau sur Exécuter l'analyse.)

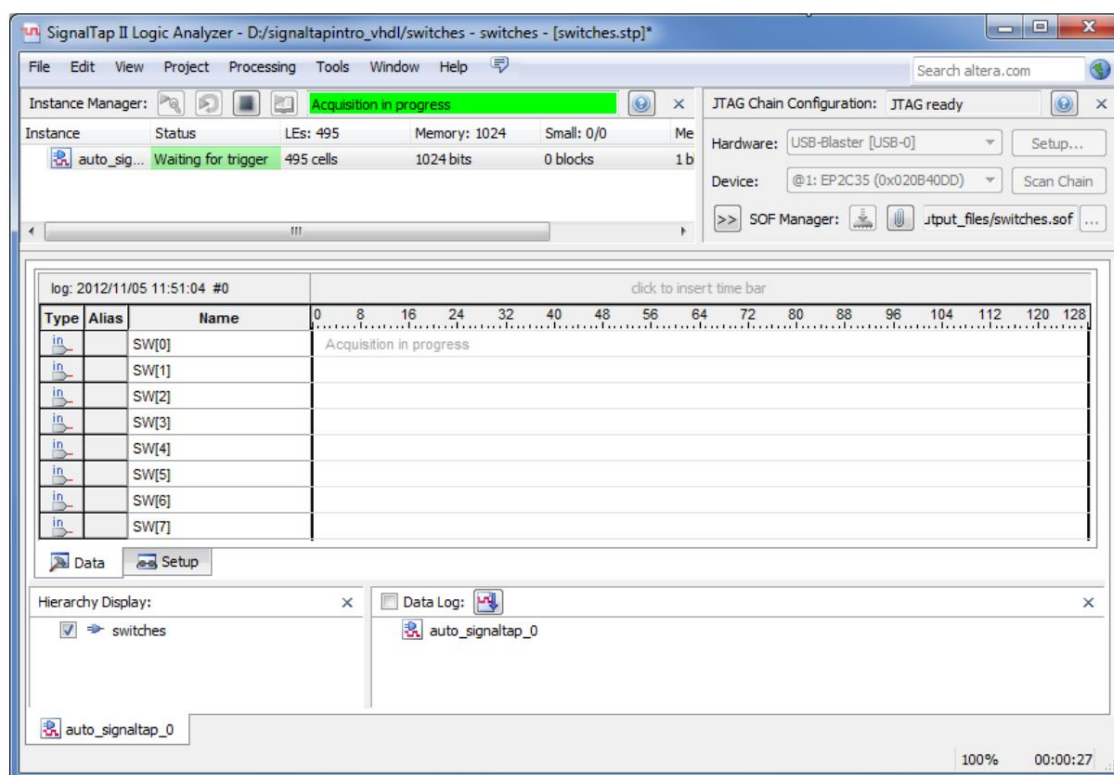


Figure 12. Fenêtre SignalTap II sur une carte de la série DE après avoir cliqué sur Exécuter l'analyse.

3. Pour observer la fonction de déclenchement de l'analyseur logique, réglez le commutateur 0 de la carte série DE sur « haut ». La fenêtre de données de SignalTap II devrait afficher l'image de la figure 13. Notez que cette fenêtre affiche les niveaux de données des 8 nœuds exploités avant et après la condition de déclenchement. Pour cela, activez l'un des commutateurs 0 à 7, puis cliquez à nouveau sur « Exécuter l'analyse ». Lorsque le commutateur 0 est à nouveau sur « haut », les valeurs des commutateurs s'affichent sur l'analyseur logique SignalTap II.

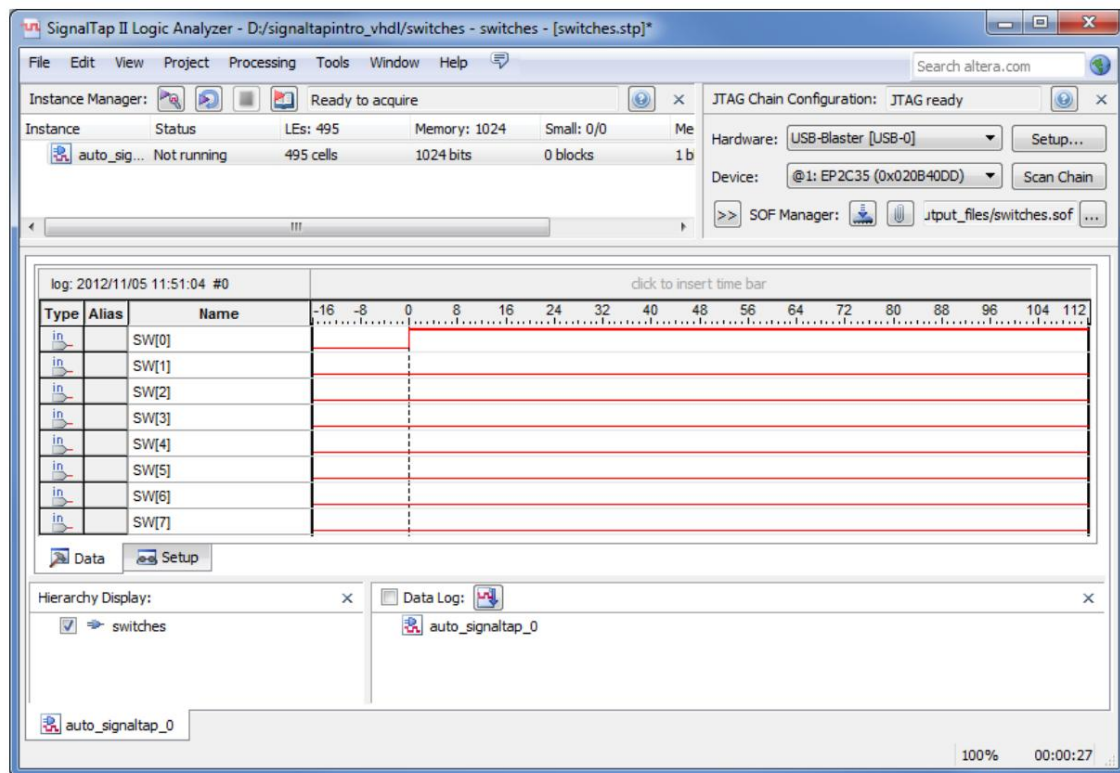


Figure 13. Affichage graphique des valeurs une fois la condition de déclenchement remplie.

7 options de déclenchement avancées

Il peut arriver que, dans une conception, vous souhaitiez une condition de déclenchement plus complexe que celle autorisée par les commandes de déclenchement de base de SignalTap. La section suivante décrit comment définir plusieurs niveaux de déclenchement.

7.1 Niveaux de déclenchement multiples

Dans cette section, nous allons configurer l'analyseur pour qu'il se déclenche lorsqu'il y a un front positif provenant du commutateur 0, du commutateur 1, du commutateur 2, puis du commutateur 3, dans cet ordre.

1. Cliquez sur l'onglet Configuration de la fenêtre SignalTap II.
2. Dans le panneau « Configuration du signal », sélectionnez « 4 » dans le menu déroulant « Conditions de déclenchement », comme illustré à la figure 14 (vous devrez peut-être faire défiler le panneau « Configuration du signal » vers le bas pour afficher ce menu). Cela modifie la liste des nœuds en créant trois nouvelles colonnes « Conditions de déclenchement ».

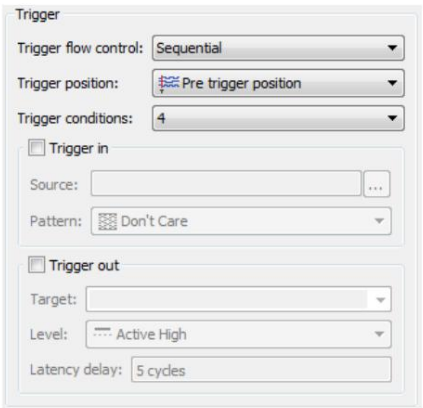


Figure 14. Réglez les niveaux de déclenchement sur 4.

3. Faites un clic droit sur la cellule « Condition de déclenchement 1 » pour SW[0] et sélectionnez « Front montant ». Faites de même pour les cellules « Condition de déclenchement 2 » pour SW[1], « Condition de déclenchement 3 » pour SW[2] et « Condition de déclenchement 4 » pour SW[3]. Vous devriez obtenir une fenêtre similaire à la figure 15.

trigger: 2012/11/01 14:22:59 #1			Lock mode: Allow all changes					
Node			Data Enable	Trigger Enable	Trigger Conditions			
Type	Alias	Name	8	8	1	2	3	4
		SW[0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
		SW[1]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
		SW[2]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
		SW[3]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
		SW[4]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
		SW[5]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
		SW[6]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
		SW[7]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				

Figure 15. Plusieurs niveaux de déclenchement définis.

4. Maintenant, recompilez la conception et chargez-la à nouveau sur la carte de la série DE.
5. Revenez à la fenêtre SignalTap II, cliquez sur l'onglet Données, puis sur Traitement > Exécuter l'analyse. Notez que la fenêtre affichera « En attente de déclenchement » jusqu'à ce que la condition de déclenchement appropriée soit remplie. Ensuite, basculez successivement vers les commutateurs 0, 1, 2, puis 3.

Une fois cette opération effectuée, vous verrez les valeurs de tous les commutateurs affichées comme dans la Figure 16. Expérimentez en suivant la procédure décrite dans cette section pour configurer d'autres conditions de déclenchement et utilisez la carte de la série DE pour tester ces conditions de déclenchement.

Pour sonder l'analyseur en continu, au lieu de cliquer sur « Exécuter l'analyse », cliquez sur « Exécuter l'analyse automatiquement », l'icône située juste à côté. Ainsi, chaque fois que la condition de déclenchement est remplie, la valeur affichée est mise à jour. Vous n'avez pas besoin de resélectionner « Exécuter l'analyse ».

Pour arrêter l'exécution automatique,

Fonction « Analyse », cliquez sur le bouton icône.

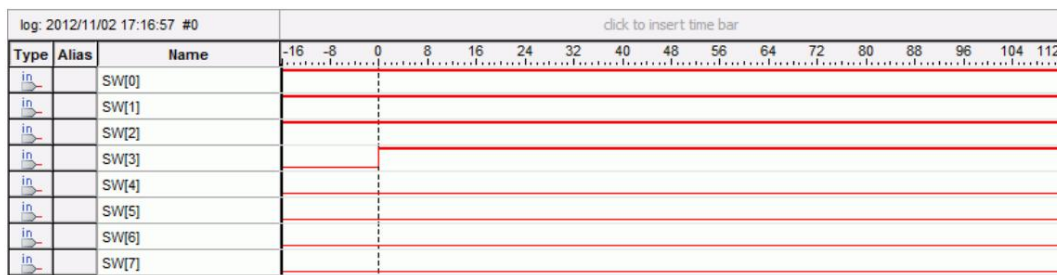


Figure 16. Affichage de l'analyseur logique lorsque les quatre conditions de déclenchement sont remplies.

7.2 Conditions de déclenchement avancées

Dans cette section, nous allons apprendre à créer des conditions de déclenchement avancées. Notre condition de déclenchement se produira dès que l'un des trois premiers affichages LED présente un front positif ou négatif. Cela signifie que l'analyseur logique actualisera son affichage à chaque modification de l'une de ces entrées. Notez que n'importe quelle fonction logique des nœuds sondés pourrait déclencher l'analyseur. Ceci n'est qu'un exemple. Après avoir implémenté cela dans les étapes suivantes, testez vos propres déclencheurs avancés.

1. Ouvrez et compilez le projet switches à partir des exemples précédents de ce didacticiel.
2. Ouvrez la fenêtre SignalTap et sélectionnez l'onglet Configuration. Dans le panneau Configuration du signal, assurez-vous que le nombre de conditions de déclenchement est défini sur 1.
3. Dans la colonne Conditions de déclenchement de la liste des nœuds, assurez-vous que la case est cochée et sélectionnez Avancé dans le menu déroulant comme dans la Figure 17. Cela fera immédiatement apparaître la fenêtre de la Figure 18. Cette fenêtre vous permet de créer un circuit logique en utilisant les différents nœuds que vous sondez avec SignalTap.

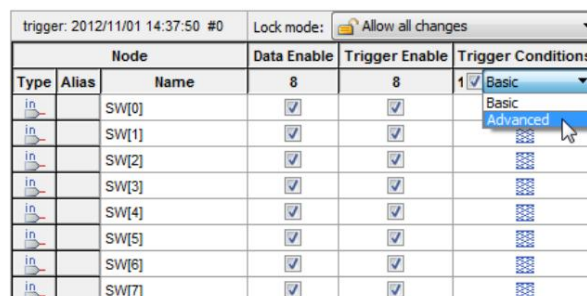


Figure 17. Sélectionnez Avancé dans le menu déroulant Niveau de déclenchement.

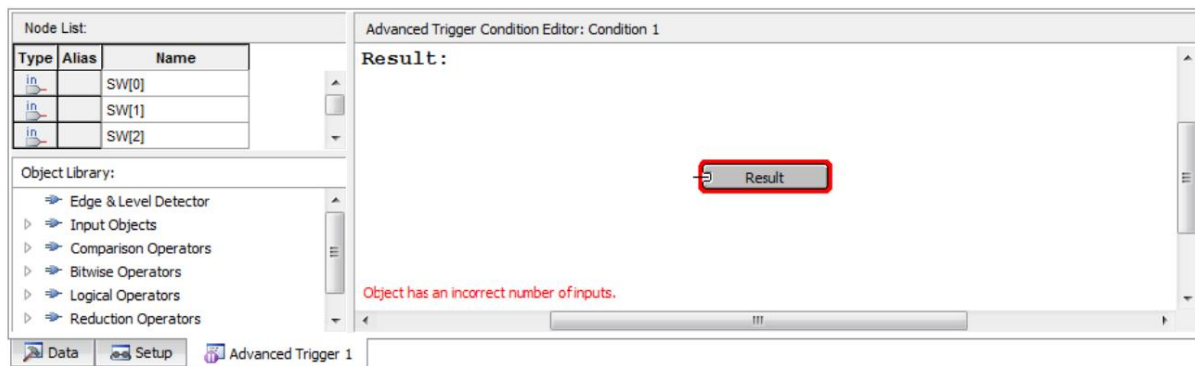


Figure 18. La fenêtre d'édition du déclencheur avancé.

4. Dans la section de liste de nœuds de cette fenêtre, mettez en surbrillance les 3 nœuds SW[0] à SW[2], puis cliquez dessus et faites-les glisser dans l'espace blanc de la fenêtre de déclenchement avancé, ce qui donne la Figure 19. Notez que vous pouvez également faire glisser et déposer chaque nœud individuellement.

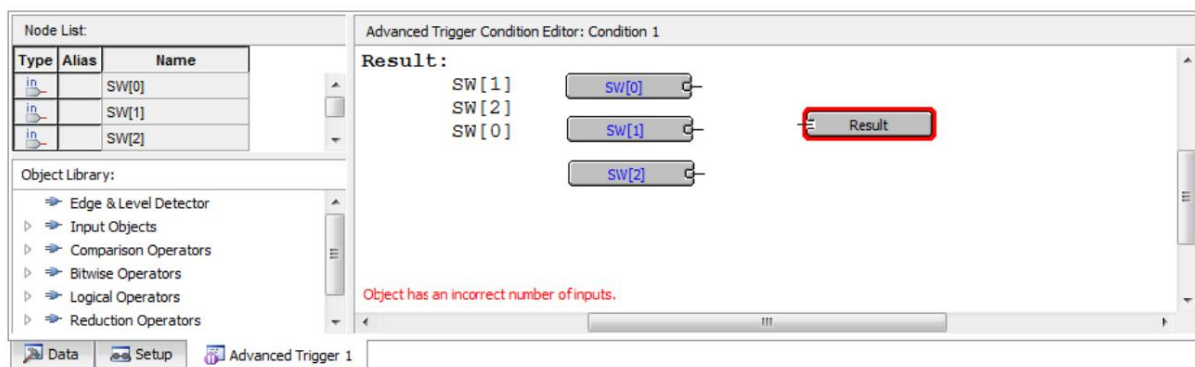


Figure 19. Les trois nœuds d'entrée d'intérêt glissés dans la fenêtre d'édition de déclencheur avancé.

5. Nous devons maintenant ajouter les opérateurs logiques nécessaires à notre circuit. Nous aurons besoin d'une porte OU ainsi que de trois détecteurs de niveau de front. Pour accéder à la porte OU, cliquez sur le signe plus à côté d'« Opérateurs logiques » dans la bibliothèque d'objets et sélectionnez « Ou logique », comme illustré à la figure 20. Glissez-déposez ensuite l'opérateur dans la fenêtre d'édition.

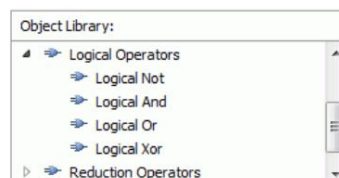


Figure 20. Sélectionnez l'opérateur Ou logique dans la fenêtre Bibliothèque d'objets et faites-le glisser dans la fenêtre d'édition.

6. Dans la bibliothèque d'objets, cliquez sur « Détecteur de bord et de niveau » et faites-le glisser dans la fenêtre d'édition. Répétez cette opération trois fois.

fois, puis disposez le circuit comme illustré à la figure 21. Les trois entrées doivent être connectées à l'entrée d'un détecteur de front et de niveau, et la sortie de chacun de ces détecteurs doit être connectée à la porte OU. La sortie de la porte OU doit être connectée à la broche de sortie déjà présente dans la fenêtre d'édition.

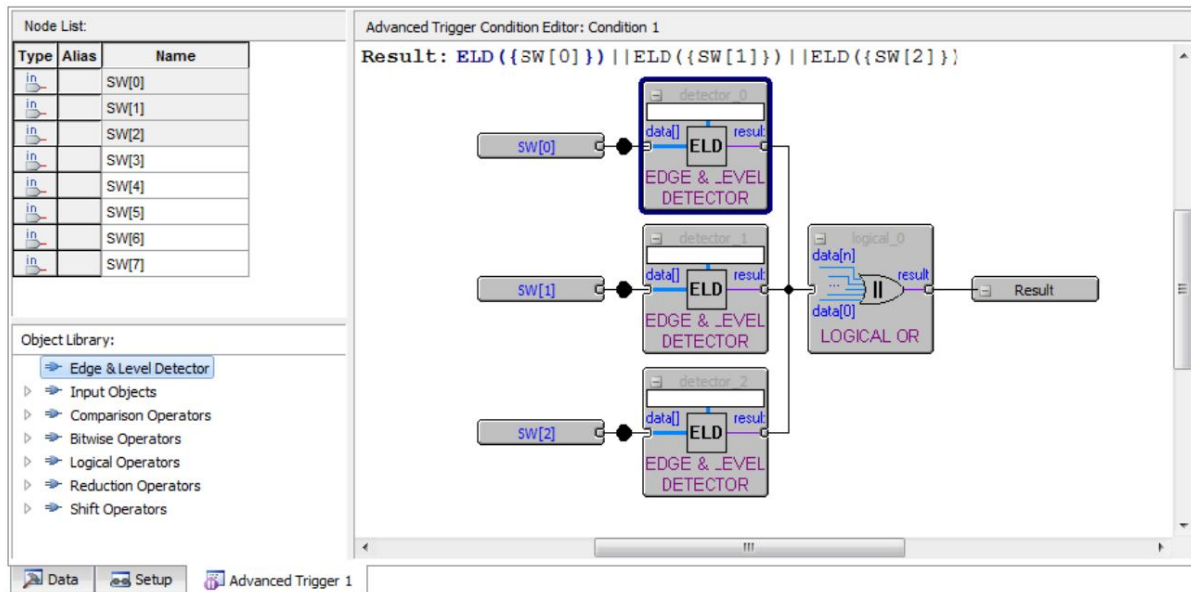


Figure 21. Disposez les éléments pour créer un circuit qui ressemble à ceci.

- Il faut maintenant configurer chaque détecteur de front et de niveau pour détecter un front descendant ou un front montant. Double-cliquez sur l'un des détecteurs de front et de niveau pour afficher la fenêtre de la figure 22. Saisissez E dans le champ de configuration, puis cliquez sur OK. Le détecteur affichera alors 1 à chaque front descendant ou montant de son entrée. Répétez cette étape pour les deux autres détecteurs de front et de niveau.

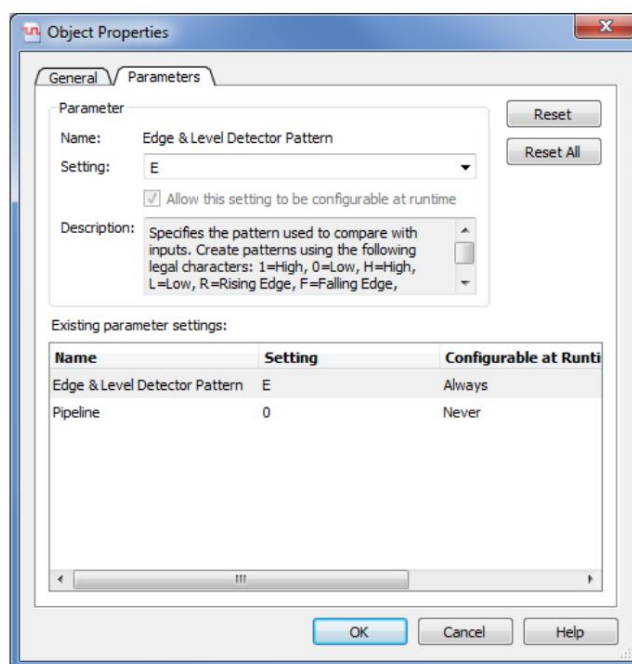


Figure 22. Tapez E dans la zone de réglage pour que la fonction se déclenche sur les fronts montants et descendants.

8. Pour tester cette condition de déclenchement avancée, compilez à nouveau le circuit conçu et chargez-le sur la carte série DE. Exécutez ensuite Signal Tap comme décrit dans la section précédente. Notez que l'analyseur doit détecter chaque fois que vous modifiez l'un des trois premiers commutateurs de la carte.

8 modes d'acquisition de profondeur d'échantillon et de tampon

Dans cette section, nous allons apprendre à définir la profondeur d'échantillonnage de notre analyseur et à utiliser les deux modes d'acquisition de tampon. Pour ce faire, nous utiliserons le projet précédent et la mise en mémoire tampon segmentée. Cette dernière permet de diviser le tampon d'acquisition en plusieurs segments distincts de taille égale. Nous créerons une profondeur d'échantillonnage de 256 bits et la diviserons en huit segments de 32 échantillons. Cela nous permettra de capturer huit événements distincts se produisant au moment du déclenchement.

1. Remettez la condition de déclenchement sur « Basic » et n'en ayez qu'une seule. Définissez la condition de déclenchement sur être à chaque bord de SW[0].
2. Dans le panneau « Configuration du signal » de la fenêtre SignalTap II, sélectionnez 256 dans le menu déroulant « Profondeur d'échantillonnage » du panneau « Données ». Cette option vous permet de spécifier le nombre d'échantillons à prélever autour des déclencheurs de votre conception. Si vous avez besoin de nombreux échantillons pour déboguer votre conception, sélectionnez une profondeur d'échantillonnage plus importante. Notez cependant qu'une profondeur d'échantillonnage trop importante risque de ne pas avoir suffisamment de place sur la carte pour votre conception et celle-ci ne sera pas compilée. Dans ce cas, essayez de réduire la profondeur d'échantillonnage.
3. Dans le panneau Configuration du signal de la fenêtre SignalTap II, dans la section Données, cochez « Segmenté ». Dans le menu déroulant à côté de « Segmenté », sélectionnez « 8 32 segments d'échantillon ». Un panneau s'affiche.

cela ressemble à la figure 23.

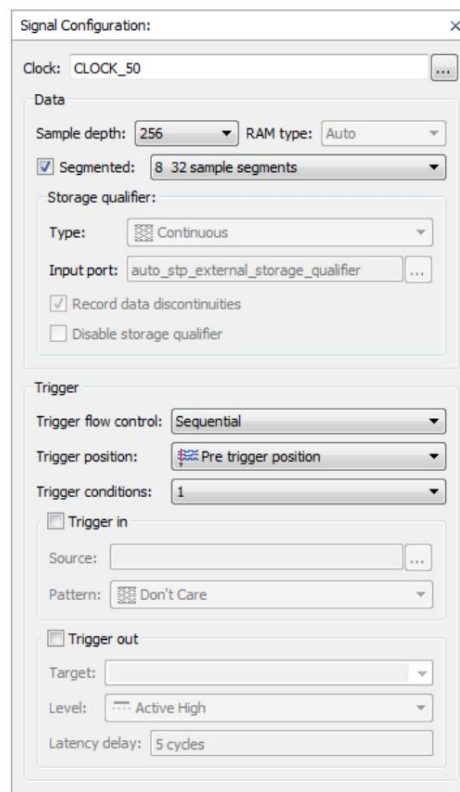


Figure 23. Sélectionnez le mode d'acquisition de tampon segmenté avec 8 segments d'échantillons de 32.

4. Recompilez et chargez le circuit conçu sur la carte de la série DE. Nous pourrions alors tester la conception.
en utilisant le mode d'acquisition segmenté.
5. Revenez à la fenêtre SignalTap II et cliquez sur Traitement > Analyse. Ensuite, basculez SW[0] vers le haut et vers le bas, et entre chaque basculement, modifiez les valeurs des 7 autres commutateurs. Après avoir répété cette opération 8 fois, les valeurs du tampon s'afficheront dans la fenêtre de données, indiquant les valeurs des 8 commutateurs à chaque déclenchement. La figure 24 présente une forme d'onde possible. Elle résulte du basculement d'un commutateur supplémentaire entre chaque basculement de SW[0].

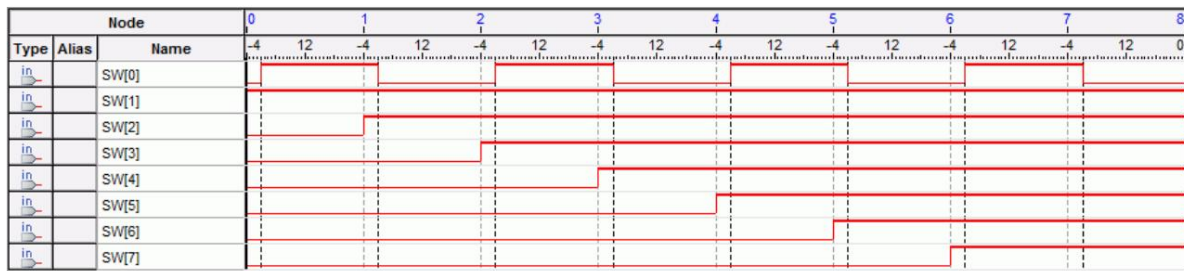


Figure 24. Formes d'onde possibles pouvant résulter de l'utilisation du mode d'acquisition segmenté.

8.1 Utilisation de l'attribut Keep

Il arrive qu'une conception que vous créez contienne des fils que le compilateur Quartus optimise. Un exemple très simple est le code VHDL ci-dessous :

```

BIBLIOTHÈQUE
ieee ; UTILISATION
ieee.std_logic_1164.all ; ENTITÉ threeInputAnd IS
    PORT ( HORLOGE_50 : DANS          STD_LOGIC;
           Sur-Quest      : DANS STD_LOGIC_VECTOR(2 JUSQU'À 0);
           LEDR            : OUT STD_LOGIC_VECTOR(0 JUSQU'À 0));
FIN troisEntréeEt;
ARCHITECTURE Comportement de trois entrées et est
    SIGNAL ab, abc : STD_LOGIC;
    ATTRIBUT garder : BOOLÉEN ;
    ATTRIBUT garder DE ab, abc : LE SIGNAL EST vrai ;
    DÉBUT ab
        <= SW(0) ET SW(1); abc <= ab ET
        SW(2);

    PROCESSUS (CLOCK_50)
        COMMENCER
            SI (RISING_EDGE(CLOCK_50)) ALORS
                LEDR(0) <= abc;
            FIN SI;
        FIN DU PROCESSUS;
    FIN Comportement ;
  
```

Figure 25. Utilisation de l'attribut « conserver » dans Quartus II.

Un schéma de ce circuit est présenté à la figure 26. Les symboles triangulaires ab et abc sont des tampons insérés par Quartus. Ils ne modifient pas les signaux qui les traversent.

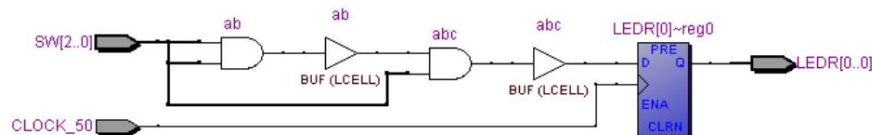


Figure 26. Le circuit implémenté par le code de la figure 25

Nous souhaitons instancier un module SignalTap II qui sondera les valeurs des entrées SW[2:0] et des sorties LEDR[2:0]. Nous souhaitons également sonder le fil interne ab. Cependant, lors de la compilation de ce code VHDL (sans les deux lignes d'attributs), le fil ab est généralement optimisé en un seul élément logique, comme illustré à la figure 27.

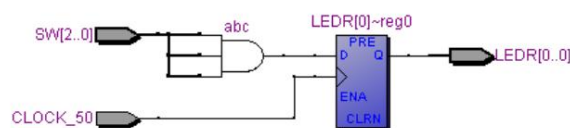


Figure 27. Le même circuit sans l'attribut « keep ».

Si vous souhaitez sonder ce fil interne, vous devrez indiquer à Quartus que vous ne souhaitez pas que ce fil soit optimisé. Pour ce faire, il faut d'abord déclarer un attribut « keep » de type booléen. C'est à cela que sert la première ligne (ATTRIBUTE keep : BOOLEAN;). Ensuite, l'attribut doit être appliqué aux signaux souhaités (ici, les signaux ab et abc). Ceci est réalisé avec la deuxième ligne (ATTRIBUTE keep OF ab, abc : SIGNAL IS true;). La figure 25 contient déjà ces lignes. Nous allons maintenant montrer comment sonder ce fil :

1. Créez un nouveau projet Quartus threeInputAnd et copiez le code VHDL de la figure 25. Compilez le projet.
2. Accédez à Outils > Analyseur logique SignalTap II, puis dans le volet Configuration de la fenêtre SignalTap II, à droite Cliquez et choisissez Ajouter des nœuds.
3. Pour le champ Filtre, sélectionnez SignalTap II : pré-synthèse. Sélectionnez [threeInputAnd] dans le menu déroulant « Regarder dans » et cliquez sur le bouton « Liste ». Déplacez les nœuds ab, SW[0], SW[1], SW[2] et LEDR[0] dans la liste « Nœuds sélectionnés », puis cliquez sur « OK ».
4. Dans le volet Configuration du signal, sélectionnez CLOCK_50 comme signal d'horloge.
5. Définissez une condition de déclenchement pour se déclencher lorsque ab devient élevé.
6. Importez le fichier d'affectation des broches correspondant à la carte série DE (ou affectez les broches manuellement, comme décrit dans la section 7 des tutoriels d'introduction de Quartus II). Pour une carte DE2, ce fichier est nommé DE2_pin_assignments.qsf.

7. Compilez à nouveau le projet.
8. Allez dans Outils > Programmeur et chargez le circuit sur la carte série DE.
9. Ouvrez à nouveau la fenêtre SignalTap et sélectionnez l'onglet Données. Placez tous les commutateurs de la carte série DE en position basse. Lancez ensuite l'analyse en sélectionnant Traitement > Exécuter l'analyse.
10. Placez les deux premiers commutateurs en position haute. La condition de déclenchement devrait être satisfaite.

Copyright ©1991-2013 Altera Corporation. Tous droits réservés. Altera, The Programmable Solutions Company, le logo Altera stylisé, les désignations spécifiques des appareils et tous les autres mots et logos identifiés comme marques commerciales et/ou marques de service sont, sauf indication contraire, les marques commerciales et marques de service d'Altera Corporation aux États-Unis et dans d'autres pays. Tous les autres noms de produits ou services appartiennent à leurs détenteurs respectifs. Les produits Altera sont protégés par de nombreux brevets et demandes en cours, droits d'auteur et droits de reproduction, américains et étrangers. Altera garantit la performance de ses semi-conducteurs conformément aux spécifications actuelles, conformément à sa garantie standard, mais se réserve le droit d'apporter des modifications à ses produits et services à tout moment et sans préavis. Altera décline toute responsabilité quant à l'application ou à l'utilisation des informations, produits ou services décrits dans le présent document, sauf accord écrit exprès d'Altera Corporation. Il est conseillé aux clients d'Altera de se procurer la dernière version des spécifications des appareils avant de se fier aux informations publiées et de passer commande de produits ou de services.

Ce document est fourni « tel quel » et à titre d'hébergement et, par conséquent, toutes les garanties, représentations ou garanties de quelque nature que ce soit (qu'elles soient expresses, implicites ou légales), y compris, sans limitation, les garanties de qualité marchande, de non-contrefaçon ou d'adéquation à un usage particulier, sont spécifiquement rejetées.