Vérification du protocole Basic VCI

Ce document présente les composants de vérification dynamique permettant de s'assurer que le protocole Basic VCI (ou BVCI) est respecté durant une simulation. Lorsque ce n'est pas le cas les erreurs identifiées sont écrites dans un fichier de log spécifié par l'utilisateur.

Cette vérification est mise en œuvre en incluant un des deux fichiers soclib\module\verification_component\bvci\caba\source\include\bvci_assert.h, soclib\module\verification_component\bvci\caba\source\include\bvci_filter.h, voire les deux fichiers pour créer les composants de vérification qui sont soit montés en série, soit sont de simples observateurs. Ces composants de conformité vérifient le protocole Basic VCI.

Les lock posés sur les adresses sont effectifs, mais ne donnent pas lieu à une vérification particulière, ce qui n'est pas le cas pour le protocole Advanced VCI.

Débogage et messages d'erreurs

Les messages d'erreur identifient le nombre de paquets traités depuis le début de la simulation et la cellule du paquet associée à l'erreur. Ces nombres sont une aide au déboguage.

Une utilisation sous gdb lorsque peu de BasicVciFilter ont été instantiés est :

```
> break BasicVciFilter::writeError
> run
> p m_nb_packets
102
> p dDirection
DOut
```

suivi de

```
> break BasicVciFilter::acquireResponseCell if m_nb_packets == 101
> run
```

Il suffit ensuite d'avancer pas à pas de manière à comprendre l'origine de l'erreur, éventuellement en activant des points d'arrêt sur d'autres processus. BasicVciFilter et BasicVciAssert sont utilisés de la même manière sous gdb.

Le format des messages d'erreur est de la forme :

```
ERROR : Protocol Error "message" on BasicVciFilter001 for the packet 102, the cell 2 issued from request !!!
ERROR : Protocol Error "message" on BasicVciAssert001 for the packet 51, the cell 1 issued from response !!!
```

où message est parmi :

- The protocol does not accept "bubble" in a VCI command request packet
  - L'erreur a lieu lorsque le signal cmdval n'est pas maintenu sur le composant initiateur alors que l'acknowledgement n'a pas eu lieu de la part du composant cible.
- The protocol does not accept "change" during request acknowledgement
  - L'erreur a lieu lorsqu'un des signaux address, be, cfixed, clen, cmd, contig, wdata, eop, cons, plen, wrap n'est pas constant sur le composant initiateur alors que l'acknowledgement n'a pas eu lieu de la part du composant cible.
- The protocol requires default request acknowledgement
  - L'erreur a lieu lorsque le signal cmdack n'est pas à true alors qu'il était positionné par défaut lorsque était arrivée la précédente requête.
- The protocol does not accept "bubble" in a VCI command response packet

    - L'erreur a lieu lorsque le signal rspval n'est pas maintenu sur le composant cible alors que l'acknowledgement de la réponse n'a pas eu lieu de la part du composant initiateur.
- The protocol does not accept "change" during response acknowledgement
    - L'erreur a lieu lorsqu'un des signaux rdata, reop, rerror n'est pas constant sur le composant cible alors que l'acknowledgement n'a pas eu lieu de la part du composant initiateur.
- The protocol requires default response acknowledgement
    - L'erreur a lieu lorsque le signal rspack n'est pas à true alors qu'il était positionné par défaut lorsque était arrivée la précédente réponse.
- Violation in the protocol : the reset command had no effect
    - L'erreur a lieu lorsque le reset n'a pas effectivement repositionné les champs cmdack et val à 0 dans la limite des 8 cycles nécessaires à cet effet. Ce nombre de 8 cycles peut être spécifié statiquement par l'intermédiaire de la méthode setDefaultReset.
- The length, command and cfixed should be constant during the reception of chain of packets
    - L'erreur a lieu lorsque un des signaux clen, cmd, cfixed n'est constant durant la réception de tous les paquets composant une chaîne.
- The fields contig, wrap, const and plen should be constant when cfixed during the reception of chain of packets
    - L'erreur a lieu lorsqu'un des signaux contig, wrap, const, plen n'est constant durant la réception de tous les paquets d'une chaîne de paquets, alors que le champ cfixed était à true pour le premier paquet de la chaîne.
- Packets should be contigous and have a length within the form 2^n
    - L'erreur a lieu lorsque le signal wrap est à true et que contig = false ou que plen n'est pas une puissance de 2 alors qu'on reçoit un nouveau paquet d'une chaîne de paquet déjà constituée.
- The length, command and cfixed should be constant during the packet reception
    - L'erreur a lieu lorsqu'un des signaux clen, cmd, cfixed n'est constant durant la reception de toutes les cellules d'un paquet.
- The fields fixed, address, contig, wrap and cons should be constant when cons during the reception of a packet
    - L'erreur a lieu lorsque un des signaux fixed, address, contig, wrap, cons n'est constant durant la reception de toutes les cellules d'un paquet, alors que le champ cons était à true pour la première cellule du paquet.
- The protocol accepts only increasing addresses in a given packet
    - L'erreur a lieu lorsque les adresses ne sont pas croissantes sur des cellules d'un même paquet alors que contig était à true pour la première cellule du paquet. Le protocole Basic VCI impose que l'adresse de la cellule suivante soit CELLSIZE + l'adresse de la cellule courante (modulo plen) et que l'adresse de la cellule courante soit alignée en conformité avec les bornes de la cellule.
- The request packet has not the expected length
    - L'erreur a lieu lorsque la longueur effective d'un paquet (le nombre de cellules consécutives associées à eop = false + 1 – pour la dernière cellule du paquet) est différente de la longueur attendue indiquée par le signal plen.
- The response packet does not correspond to any request
    - L'erreur a lieu lorsqu'une réponse arrive alors qu'aucune requête ne lui correspond, ce qui se produit lorsque plus de réponses ont été traitées que de requêtes.
- The response packet has not the expected length
    - L'erreur a lieu lorsqu'une réponse arrive alors que la requête qui devrait lui correspondre ne lui correspond pas, en particulier sur le critère de la longueur. La raison est soit un problème de correspondance, soit le fait que le paquet-réponse soit composé de trop ou pas assez de cellule.

Le composant BasicVciFilter

Ce composant de vérification est monté en série dans une architecture existante bâtie autour du protocole Basic VCI. Il capture ses données en fin de front descendant après que toutes les méthodes genMoore des composants aient terminé leur travail et notre composant effectue la vérification durant le front montant.

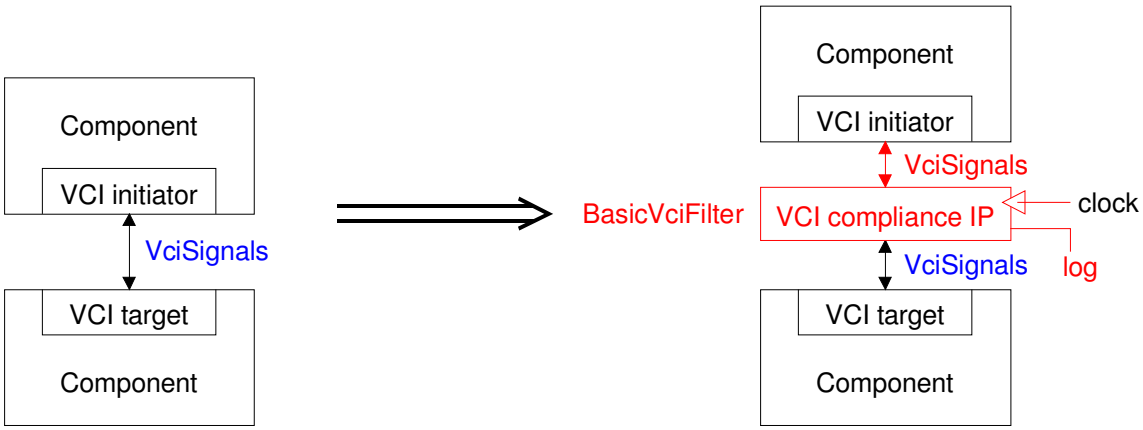

Figure 1 : composant de vérification monté en série

L'horloge de BasicVciFilter n'est pas forcément globale mais doit être commune avec le composant initiateur et le composant cible.

L'interface caba de ces composants est la suivante. Lorsqu'un utilisateur souhaite introduire un composant de vérification sur un signal de type Basic VCI, il introduit à l'élaboration le code en rouge et en italique (sur la partie droite). La vérification est alors automatique.

```
template <typename vci_param>
class BasicVciFilter
        : public soclib::caba::BaseModule {
 public:
  struct In {
    In(const std::string &name);
    void operator()(VciSignals<vci_param> &sig);
  };

  struct Out {
    Out(const std::string &name);
    void operator()(VciSignals<vci_param> &sig);
  };

  sc_in<bool> p_clk;
  In p_in;
  Out p_out;

 protected:
  SC_HAS_PROCESS(BasicVciFilter);

 public:
  BasicVciFilter(sc_module_name insname);
  void reset(); // process sensitive to p_resetn.pos()
  void copy(); // process that copies inputs to outputs
  void transition(); // process that does
        // the verification on the rising edge of p_clk

   // configuration methods
    // set max cycle number before reset is effective
  void setDefaultReset(int uDefaultReset);
    // set the error file
  void setLogOut(std::ostream& out);
    // assume that Basic VCI is in default mode for be
  void setDefaultMode();
    // assume that Basic VCI is in free mode for be
  void setFreeMode();
};
```

```
// sc_main implementation

VciFstComponent<MyVciParams> vciFst("VciFstComponent");
VciSndComponent<MyVciParams> vciSnd("VciSndComponent");
sc_clock clk("Clock", 1, 0.5, 0.0);
soclib::caba::VciSignals<MyVciParams> vciSignals("VciSignals");

#ifdef SOCLIB_VERIF
std::ofstream log_file("verif.log");
soclib::caba::VciSignals<MyVciParams>
                                vciSignalsVerif("VciSignals_verif");
soclib::caba::BasicVciFilter<MyVciParams> vciFilter("VciFilter_verif");
vciFilter.p_clk(clk);
vciFilter.setLogOut(log_file);
vciFilter.p_in(vciSignals);
vciFilter.p_out(vciSignalsVerif);
// optional
vciFilter.setDefaultResult(8);
vciFilter.setDefaultMode();
#else
  VciSignals<MyVciParams>& vciSignalsVerif = vciSignals;
#endif

vciFst.p_clk(clk);
vciFst.p_vci(vciSignals);
vciSnd.p_clk(clk);
vciSnd.p_vci(vciSignalsVerif);
```

Le composant BasicVciAssert

Ce composant de vérification se contente d'observer le contenu d'un VciSignals sur chaque front montant d'horloge suivant le schéma de la Figure 2. Du point de vue fonctionnel il effectue les mêmes vérifications que BasicVciFilter, mais il est moins intrusif.

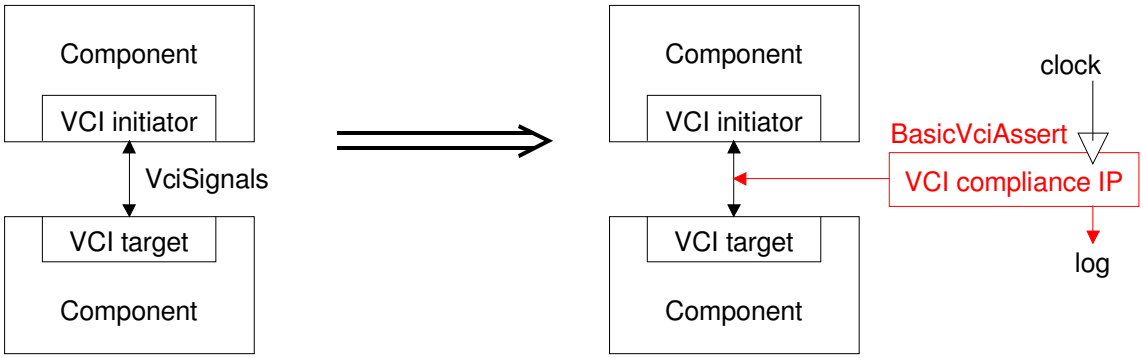

Figure 2 : composant de vérification observateur

Les composants BasicVciAssert sont garantis ne pas perturber le comportement du système. En particulier, ils n'ont pas de capacité de filtrer. La vérification de la cohérence a lieu sur chaque front montant d'horloge. Ces composants n'ont qu'une sortie qui est le fichier de log pour enregistrer les erreurs détectées.

L'horloge de BasicVciAssert, comme celle de BasicVciFilter, n'est pas forcément globale mais doit être commune avec le composant initiateur et le composant cible, afin de ne pas manquer des évènements sur les entrées.

L'interface caba de ces composants est la suivante. Lorsqu'un utilisateur souhaite introduire un composant de vérification sur un signal de type Basic VCI, il introduit à l'élaboration le code en rouge et en italique (sur la partie droite). La vérification est alors automatique.

```
template <typename vci_param>
class BasicVciAssert
                : public soclib::caba::BaseModule {
 public:
  VciSignals<vci_param>& observedSignals;
  sc_in<bool> p_clk;

 protected:
  SC_HAS_PROCESS(BasicVciAssert);

 public:
  BasicVciAssert(VciSignals<vci_param>& ref,
     sc_module_name insname);

  void reset(); // process sensitive to p_resetn.pos()
  void transition(); // process that does
          // the verification on the rising edge of p_clk

   // configuration methods
     // set max cycle number before reset is effective
  void setDefaultReset(int uDefaultReset);
     // set the error file
  void setLogOut(std::ostream& out);
     // assume that Basic VCI is in default mode for be
  void setDefaultMode();
     // assume that Basic VCI is in free mode for be
  void setFreeMode();
};
```

```
// sc_main implementation

VciFstComponent<MyVciParams> vciFst("VciFstComponent");
VciSndComponent<MyVciParams> vciSnd("VciSndComponent");
sc_clock clk("Clock", 1, 0.5, 0.0);
soclib::caba::VciSignals<MyVciParams> vciSignals("VciSignals");

#ifdef SOCLIB_VERIF
std:auto_ptr<BasicVciAssert<MyVciParams> > pvciAssert;
std::ofstream log_file("verif.log");
pvciAssert = new PVciAssert<MyVciParams>(vciSignals, "vciAssert");
pvciAssert->p_clk(clk);
pvciAssert->setLogOut(log_file);
// optional
pvciAssert->setDefaultResult(8);
pvciAssert->setDefaultMode();
#endif

vciFst.p_clk(clk);
vciFst.p_vci(vciSignals);
vciSnd.p_clk(clk);
vciSnd.p_vci(vciSignal);
```

Conclusion

L'introduction de composants de vérification a été mise en place pour ne pas perturber le système. Néanmoins elle ralentit les temps de simulation. Il est donc important de pouvoir la désactiver ou de ne vérifier le protocole Basic VCI qu'entre certains composants. Cette désactivation est mise en œuvre :

- soit en plaçant le code de vérification sous condition que la macro SOCLIB_VERIF est définie et en proposant plusieurs modes de compilation pour le système : un mode avec g++ -O2 … et un mode avec g++ -DSOCLIB_VERIF.
- soit en ne créant les composants de vérification que si spécifié par l'utilisateur sur la ligne de commande.