Multiple Angio Detector Synchronization

Germán Vera

Rev 1 – September 17, 2010: first version
Rev 2 – September 30, 2010: Added description of alternative method of synchronization, fixed errors in example of method A and added example for method B.
Rev 3 – February 01, 2011: simplified example B, added comments, and added simple time diagram

imagination at work

GE Proprietary Information – Do not distribute without prior authorization

Synchronization Description

The DFN/DAB(*) has real-time lines that can be configured as input or outputs. By default, ADEPT Core configures channels 0 to 5 as output and channels 6 & 7 as input. The status of the real-time lines configured as outputs can be controlled by the Flag instruction in a DFN/DAB script. The Wait instruction in a DFN/DAB script can be used to wait until a specific real-time line status is detected by the DFN/DAB.

To achieve readout synchronization, the slave PCs would start running DFN/DAB scripts that use the Wait instruction to hold script execution until a real-time signal status is received. Then, the master PC will start running a DFN/DAB script that generates the expected real-time line status for the slave DFNs/ DABs. After the signal is issued/detected, all DFN/DAB scripts will perform a number of scrubs at the desired time between frames to stabilize the offset images from all detectors. Then, image acquisition can start. The accuracy of the synchronization start will be within +- 2us.

After the initial synchronization, the scripts running in the DFN/DAB boards could:

A. Run independently without additional synchronization

B. Be synchronized periodically

Tests of the first method of synchronization using DAB boards showed a drift between 0.2 and 0.5us per frame (when running at 30fps Angio TM0) after 1000 frames. Tests with DFN boards showed no drift.

Tests of the second method of synchronization showed an accuracy within 2us per frame.

Reminder: the DFN/DAB scripts needed for synchronized acquisition are not generated by Adept Core. They need to be created outside of ADEPT Core and then they can be run as a “User sequence”.

(*) DAB: new version of the DFN board

GE Proprietary Information – Do not distribute without prior authorization

Synchronization Option A DFN/DAB Script Example – Master

```
sub test_frame
{
   $frame_name = 'test_frame';
   %qv = ('del_qv10'  => [4997],
          'si_cmd'  => [0x1002],
          'si_data'  => [0]);
   %qp = ();
   compile_init(@_, \%qp, \%qv, $frame_name);

   #Master script
   Flag(0x001000);          #RTB Ch#4 ON
   LoopN_begin(4);
   Send(0x0,0);
   Delay(11864);            #RTR + FRT
   Delay('del_qv10');       #DEL_QV10 = TBF
   LoopN_end();
   Send(0x0,0);
   Delay(11864);            #RTR + FRT

   LoopN_begin(5);
   Delay('del_qv10');       #DEL_QV10 = TBF
   Send(0x800000,0);
   Delay(11816);            #FRT + RTR
   LoopN_end();

   Flag(0xFF0F00);          #RT_EXPOSE_OFF
   Delay(4997);             #TBF

   compile_finit();
}
```

Generate RT signal to sync with slave DFN/DABs and perform 5 scrubs to get to stable offsets
Acquisition timing controlled by Master script

Acquire 5 images

A script generated by ADEPT Core was modified/simplified to *generate* a real-time signal for synchronization and perform scrubs to stabilize offset images (instructions in light blue). The actual RT line used for synchronization can be programmed.

GE Proprietary Information – Do not distribute without prior authorization

Synchronization Option A DFN/DAB Script Example – Slave

```
sub test_frame
{
   $frame_name = 'test_frame';
   %qv = ('del_qv10'  => [4997],
          'si_cmd'  => [0x1002],
          'si_data'  => [0]);
   %qp = ();
   compile_init(@_, \%qp, \%qv, $frame_name);

   #Slave script
   Wait(0x004000);          #Wait for RTB Ch#6 ON
   LoopN_begin(4);
   Send(0x0,0);
   Delay(11864);            #RTR + FRT
   Delay('del_qv10');       #DEL_QV10 = TBF
   LoopN_end();
   Send(0x0,0);
   Delay(11864);            #RTR + FRT

   LoopN_begin(5);
   Delay('del_qv10');       #DEL_QV10 = TBF
   Send(0x800000,0);
   Delay(11816);            #FRT + RTR
   LoopN_end();

   Flag(0xFF0F00);          #RT_EXPOSE_OFF
   Delay(4997);             #TBF

   compile_finit();
}
```

Wait for RT signal from Master DFN/DAB and perform 5 scrubs to get to stable offsets
Acquisition timing controlled by Slave script

Acquire 5 images

A script generated by ADEPT Core was modified/simplified to *wait* for the real-time signal and perform scrubs to stabilize offset images (instructions in light blue). The actual RT line used for synchronization can be programmed.

GE Proprietary Information – Do not distribute without prior authorization

Synchronization Option B DFN/DAB Script Example – Master

```
sub test_frame
{
   $frame_name = 'test_frame';
   %qv = ('del_qv10'  => [4997],
          'si_cmd'  => [0x1002],
          'si_data'  => [0]);
   %qp = ();
   compile_init(@_, \%qp, \%qv, $frame_name);

   #Master script
   LoopN_begin(4);
   Flag(0x001000);        #RTB Ch#4 ON
   Delay(0);              #4 us delay
   Flag(0xFF1000);        #RTB Ch#4 OFF
   Send(0x0,0);           #Scrub cmd. Takes 2us to execute
   Delay(11864);          #RTR + FRT
   Delay('del_qv10');     #DEL_QV10 = TBF
   LoopN_end();
   LoopN_begin(3);
   Flag(0x001000);        #RTB Ch#4 ON
   Delay(0);              #4 us delay
   Flag(0xFF1000);        #RTB Ch#4 OFF
   Send(0x800000,0);      #Readout cmd. Takes RTR to execute
   Delay(11816);          #FRT
   Delay('del_qv10');     #DEL_QV10 = TBF
   LoopN_end();

   compile_finit();
}
```

Generate RT signal per frame to sync with slave DFN/DABs and perform 5 scrubs to get to stable offsets
Acquisition timing controlled by Master script

These delays need to be set according to the firmware (rad/angio), scan mode (bin/roi for angio) and timing mode

Generate RT signal per frame and acquire 5 images

GE Proprietary Information – Do not distribute without prior authorization

Synchronization Option B DFN/DAB Script Example – Slave

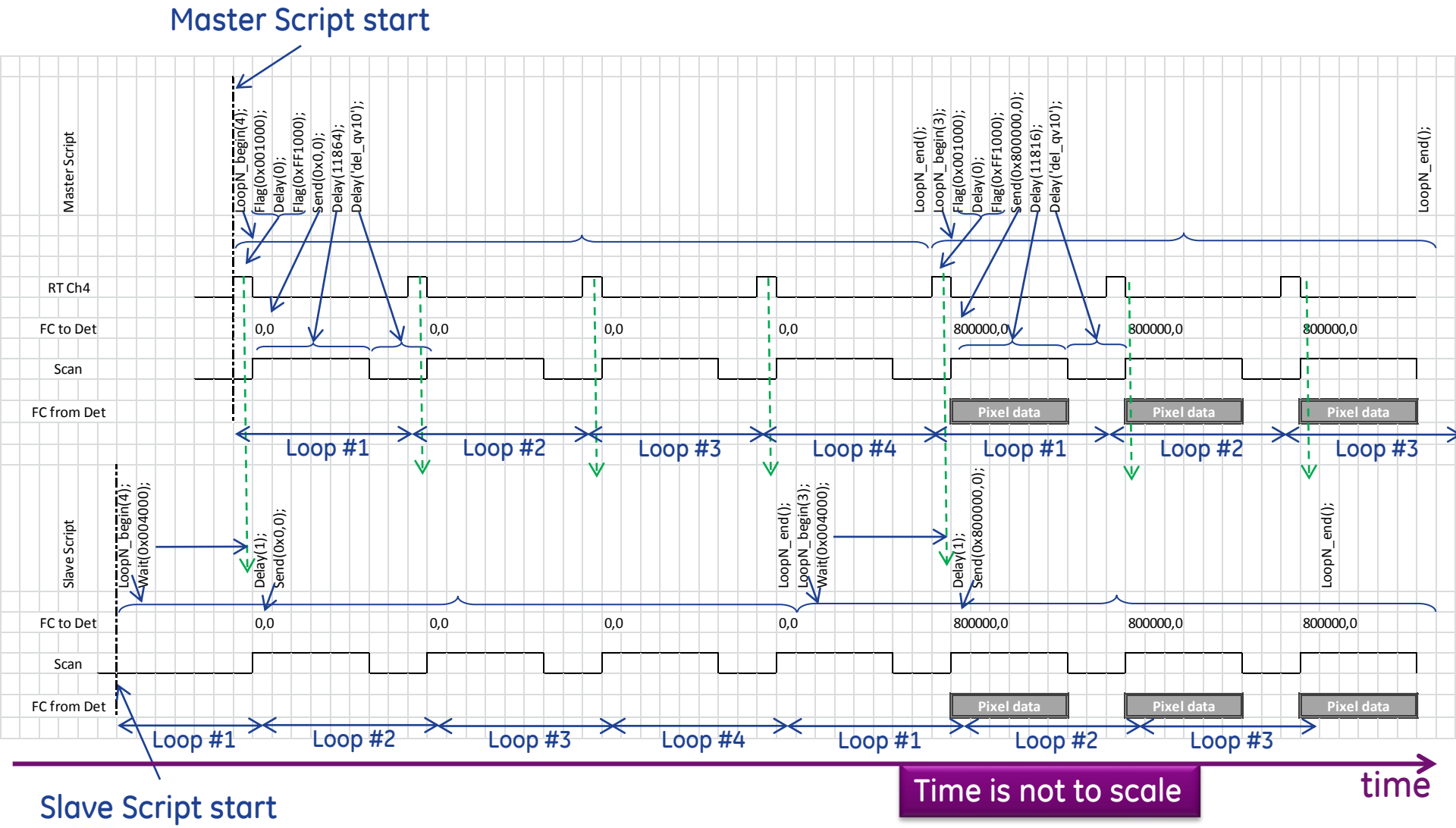
```
sub test_frame
{
   $frame_name = 'test_frame';
   %qv = ('del_qv10'  => [4997],
          'si_cmd'  => [0x1002],
          'si_data'  => [0]);
   %qp = ();
   compile_init(@_, \%qp, \%qv, $frame_name);

   #Slave script
   LoopN_begin(4);
   Wait(0x004000);          #Wait for RTB Ch#6 ON
   Delay(1);                #6 us delay
   Send(0x0,0);             #Takes 2us to execute

   LoopN_end();
   LoopN_begin(5);
   Wait(0x004000);          #Wait for RTB Ch#6 ON
   Delay(1);                #6 us delay
   Send(0x800000,0);        #Takes RTR to execute
   LoopN_end();

   compile_finit();
}
```

Wait for RT signal per frame to sync with master DFN/DAB and perform 5 scrubs to get to stable offsets
Acquisition timing controlled by Master script

Generate RT signal per frame and acquire 5 images

GE Proprietary Information – Do not distribute without prior authorization

Script Execution Timing – Option B



imagination at work

GE Proprietary Information – Do not distribute without prior authorization

DFN Primitives

The following primitives can be used to construct DFN scripts:

- **Send(Hdr1, Hdr2):** Transmit FC command to the detector. Hdr1 & Hdr2 are 32bit long. Execution time depends on when a reply from the detector is received (2us when detector is not scanning), or timeout set (if no reply is received).
- **Delay(T):** Wait (T+2)*2us. T is 32 bit long.
- **Flag(F):** F is 3 bytes to control 8 bit signal on output port (host or RT bus). Execution time is 2us.
- **LoopKF(K,F):** Repeat last K lines until flag F is received (from host or RT). To use this primitive, the LoopKF_begin(N) and LoopKF_end() functions can be used in the Perl script. Execution time is 2us.
- **Wait(F):** Pause the execution of the queue until flag F is received (from host or RT bus). Minimum execution time is 2us.
- **LoopKN(K,N):** Repeat last K lines N+2 times. To use this primitive, the LoopN_begin(N) and LoopN_end() functions can be used in the Perl script. Execution time is 2us.

GE Proprietary Information – Do not distribute without prior authorization