Dynamically Configured Java Based Register Windows For Efficient Simulation Debug

Dan Helm

ARM Inc.

September 23, Year 2014

SNUG Austin

Agenda

Introduction

Verdi Register Window

Novas Programming Interface

Dynamic Register GUI

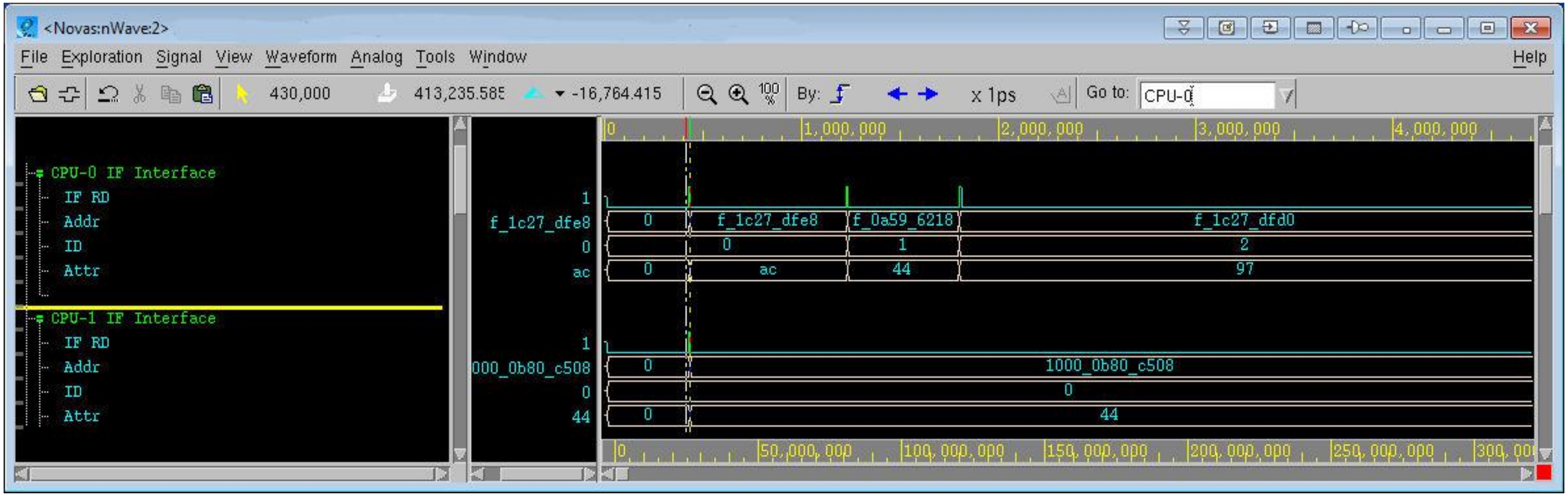
Conclusions

Introduction

- *Dynamically Configured Java Based Register Windows For Efficient Simulation Debug*
- Configurable Systems
- Consistent Debug View
  - Active Debugger
  - Logs
  - Wave Dumps
  - Verdi Register Windows

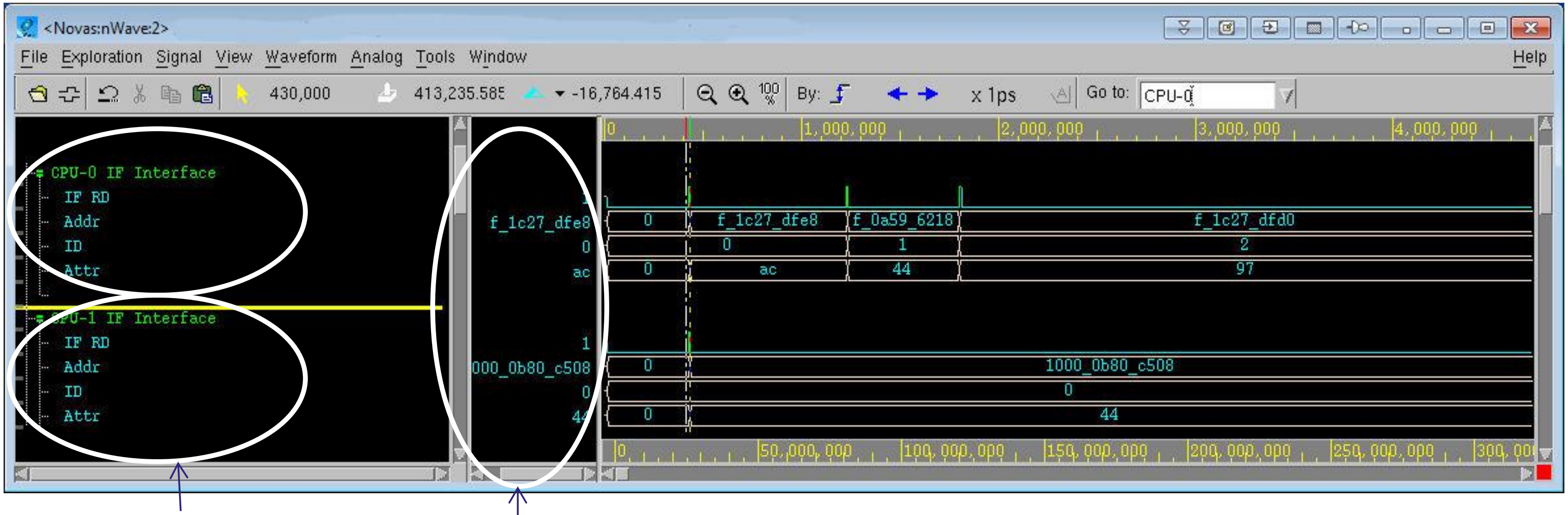
Verdi Register Window

nWave Window



Verdi Register Window

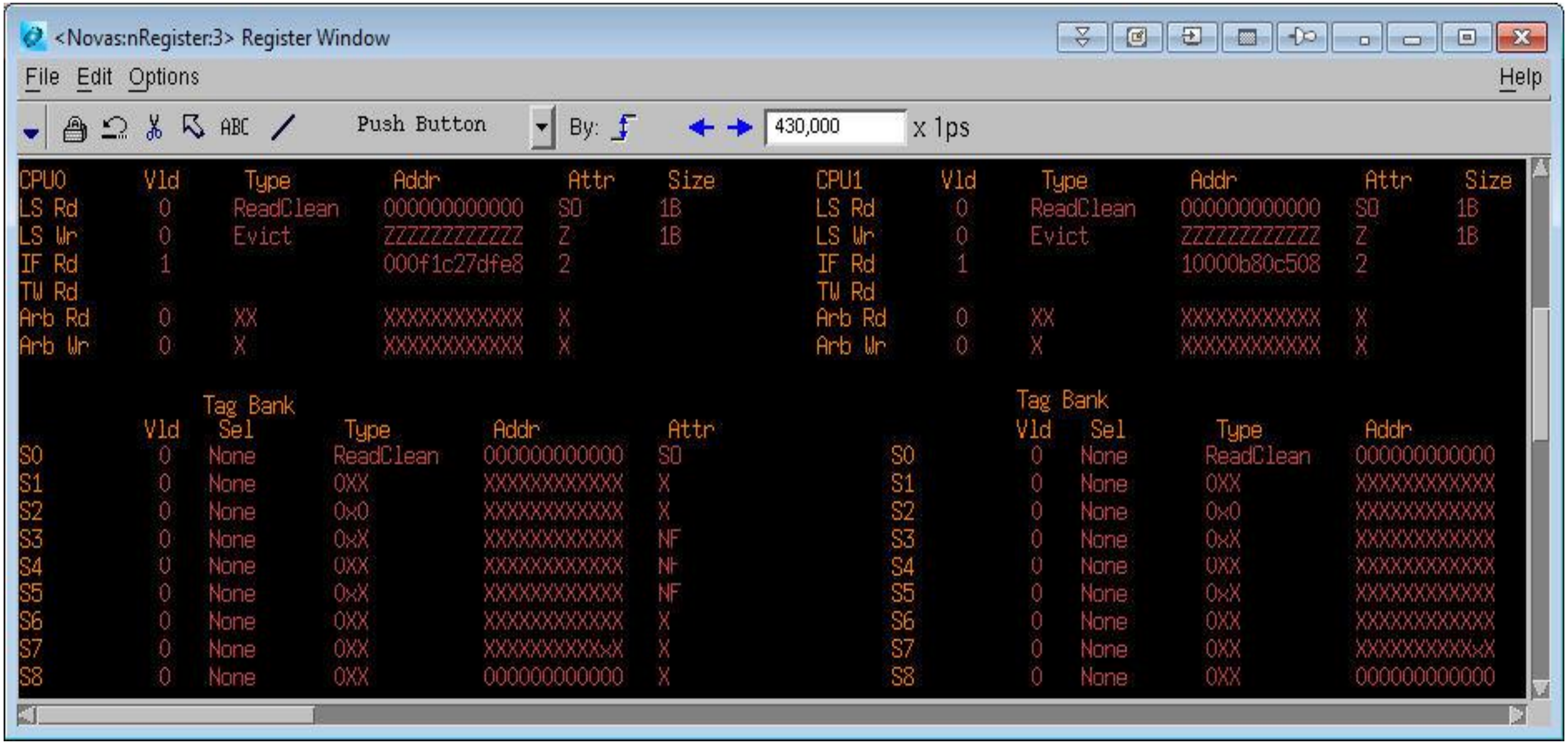
nWave Window



Fetch Interfaces for two CPUs going to a larger cache

No relationship information shown

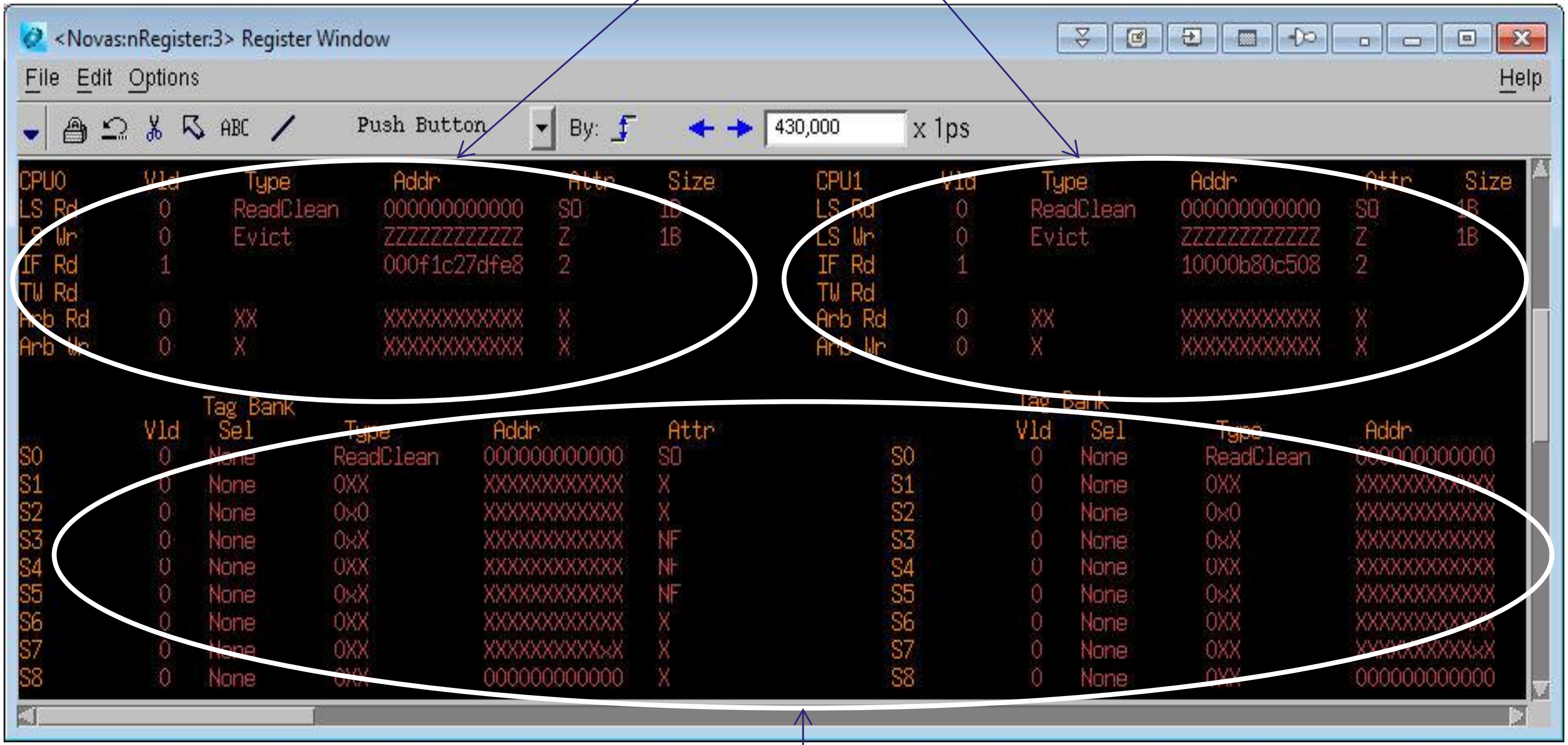
Verdi Register Window



```
CPU0      Vld    Type       Addr          Attr   Size       CPU1      Vld    Type       Addr          Attr   Size
LS Rd      0     ReadClean  000000000000  S0     1B         LS Rd      0     ReadClean  000000000000  S0     1B
LS Wr      0     Evict      ZZZZZZZZZZZZ  Z      1B         LS Wr      0     Evict      ZZZZZZZZZZZZ  Z      1B
IF Rd      1                000f1c27dfe8  2                 IF Rd      1                10000b80c508  2
TW Rd                                                       TW Rd
Arb Rd     0     XX         XXXXXXXXXXXX  X                 Arb Rd     0     XX         XXXXXXXXXXXX  X
Arb Wr     0     X          XXXXXXXXXXXX  X                 Arb Wr     0     X          XXXXXXXXXXXX  X

               Tag Bank                                                   Tag Bank
          Vld   Sel        Type        Addr          Attr                 Vld   Sel        Type        Addr
S0         0    None       ReadClean   000000000000  S0            S0      0    None       ReadClean   000000000000
S1         0    None       0XX         XXXXXXXXXXXX  X             S1      0    None       0XX         XXXXXXXXXXXX
S2         0    None       0x0         XXXXXXXXXXXX  X             S2      0    None       0x0         XXXXXXXXXXXX
S3         0    None       0xX         XXXXXXXXXXXX  NF            S3      0    None       0xX         XXXXXXXXXXXX
S4         0    None       0XX         XXXXXXXXXXXX  NF            S4      0    None       0XX         XXXXXXXXXXXX
S5         0    None       0xX         XXXXXXXXXXXX  NF            S5      0    None       0xX         XXXXXXXXXXXX
S6         0    None       0XX         XXXXXXXXXXXX  X             S6      0    None       0XX         XXXXXXXXXXXX
S7         0    None       0XX         XXXXXXXXXXxX  X             S7      0    None       0XX         XXXXXXXXXXxX
S8         0    None       0XX         000000000000  X             S8      0    None       0XX         000000000000
```

Verdi Register Window

Interfaces to cache for two CPUs



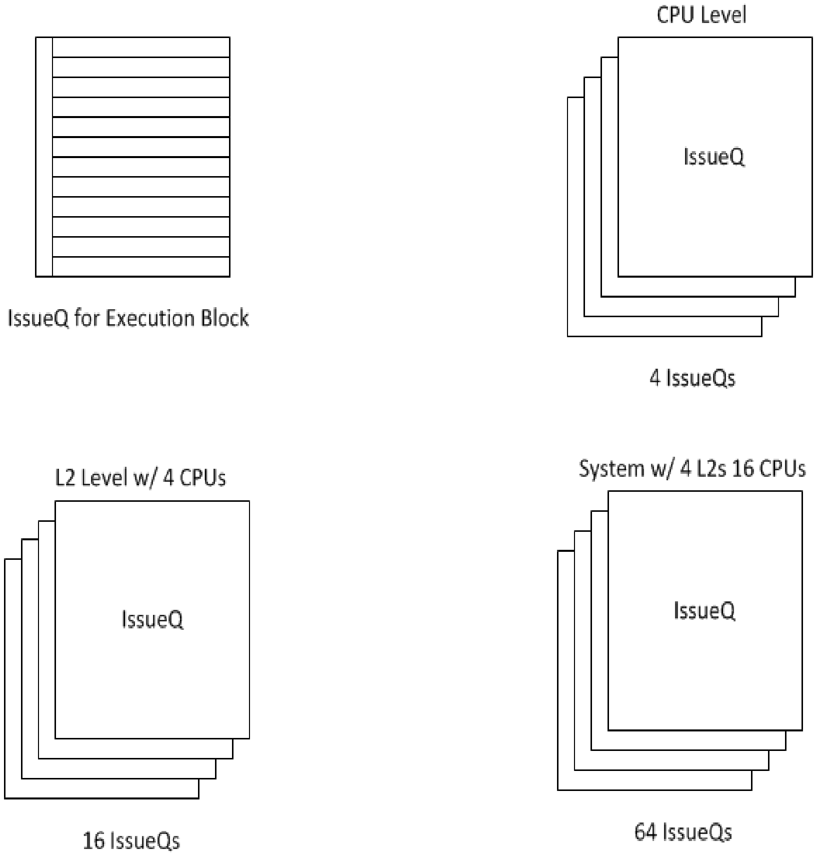
```
CPU0      Vld   Type       Addr          Attr   Size      CPU1      Vld   Type       Addr          Attr   Size
LS Rd     0     ReadClean  000000000000  S0     1B        LS Rd     0     ReadClean  000000000000  S0     1B
LS Wr     0     Evict      ZZZZZZZZZZZZ  Z      1B        LS Wr     0     Evict      ZZZZZZZZZZZZ  Z      1B
IF Rd     1                000f1c27dfe8  2                IF Rd     1                10000b80c508  2
TW Rd                                                     TW Rd
Arb Rd    0     XX         XXXXXXXXXXXX  X                Arb Rd    0     XX         XXXXXXXXXXXX  X
Arb Wr    0     X          XXXXXXXXXXXX  X                Arb Wr    0     X          XXXXXXXXXXXX  X

             Tag Bank                                                       Tag Bank
          Vld   Sel     Type       Addr          Attr                  Vld   Sel     Type       Addr
S0        0     None    ReadClean  000000000000  S0             S0     0     None    ReadClean  000000000000
S1        0     None    0XX        XXXXXXXXXXXX  X              S1     0     None    0XX        XXXXXXXXXXXX
S2        0     None    0x0        XXXXXXXXXXXX  X              S2     0     None    0x0        XXXXXXXXXXXX
S3        0     None    0xX        XXXXXXXXXXXX  NF             S3     0     None    0xX        XXXXXXXXXXXX
S4        0     None    0XX        XXXXXXXXXXXX  NF             S4     0     None    0XX        XXXXXXXXXXXX
S5        0     None    0xX        XXXXXXXXXXXX  NF             S5     0     None    0xX        XXXXXXXXXXXX
S6        0     None    0XX        XXXXXXXXXXXX  X              S6     0     None    0XX        XXXXXXXXXXXX
S7        0     None    0XX        XXXXXXXXXXxX  X              S7     0     None    0XX        XXXXXXXXXXxX
S8        0     None    0XX        000000000000  X              S8     0     None    0XX        000000000000
```

Pipeline information for two banks

Verdi Register Window

nRegister Limitations

- No Multiple Instantiations
- File Management
- Limited Graphical Options
- Single data source

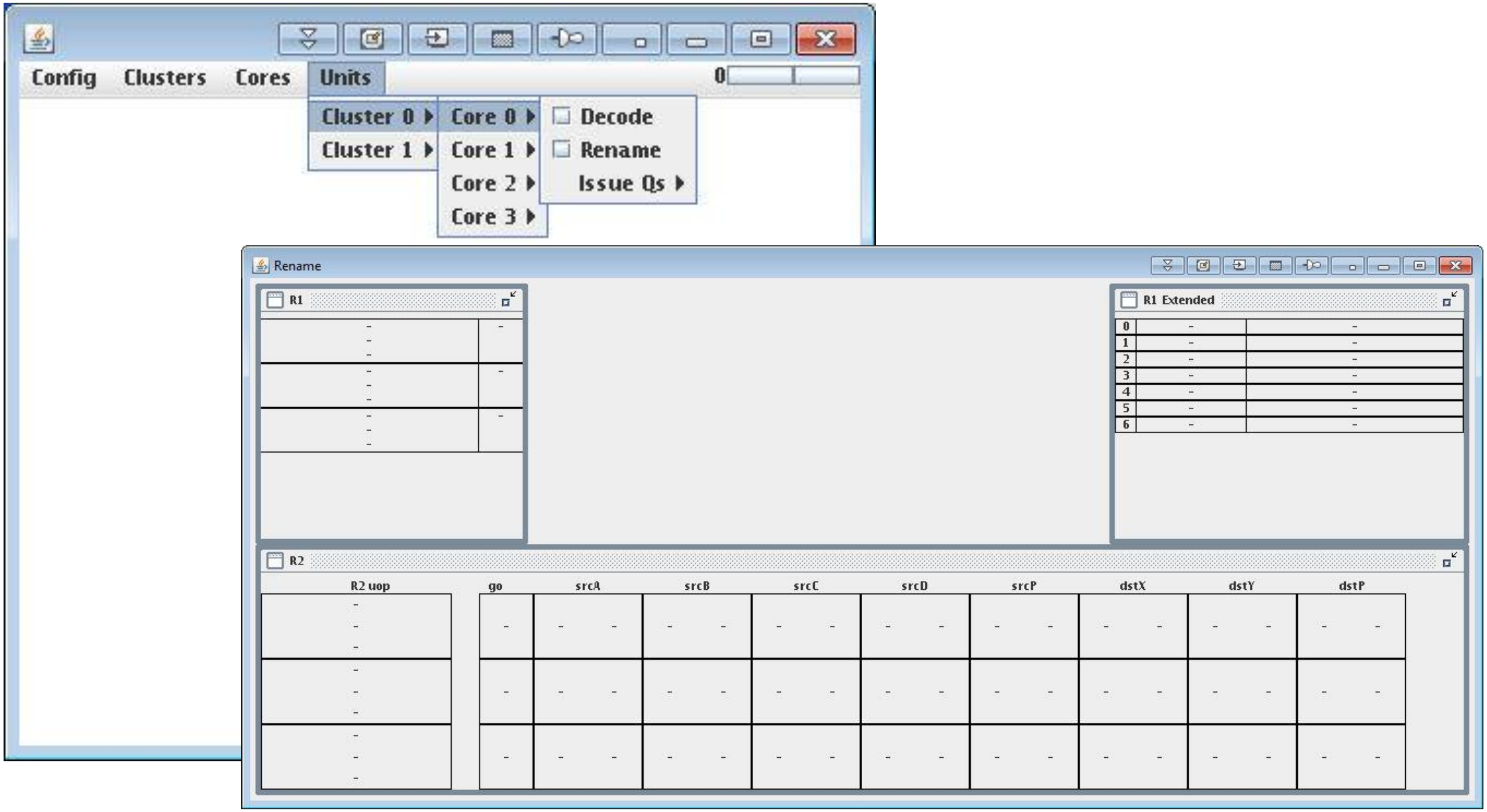


NPI

Novas Programming Interface

- Group of Tcl and C APIs to manipulate FSDBs and Verdi
- Functionality
  - Retrieve data values
  - Traverse Design
  - Traverse Time
  - Manipulate Verdi Windows
  - See Verdi Window Changes

Dynamic Register GUI



R1

- - -	-
- - -	-
- - -	-

R1 Extended

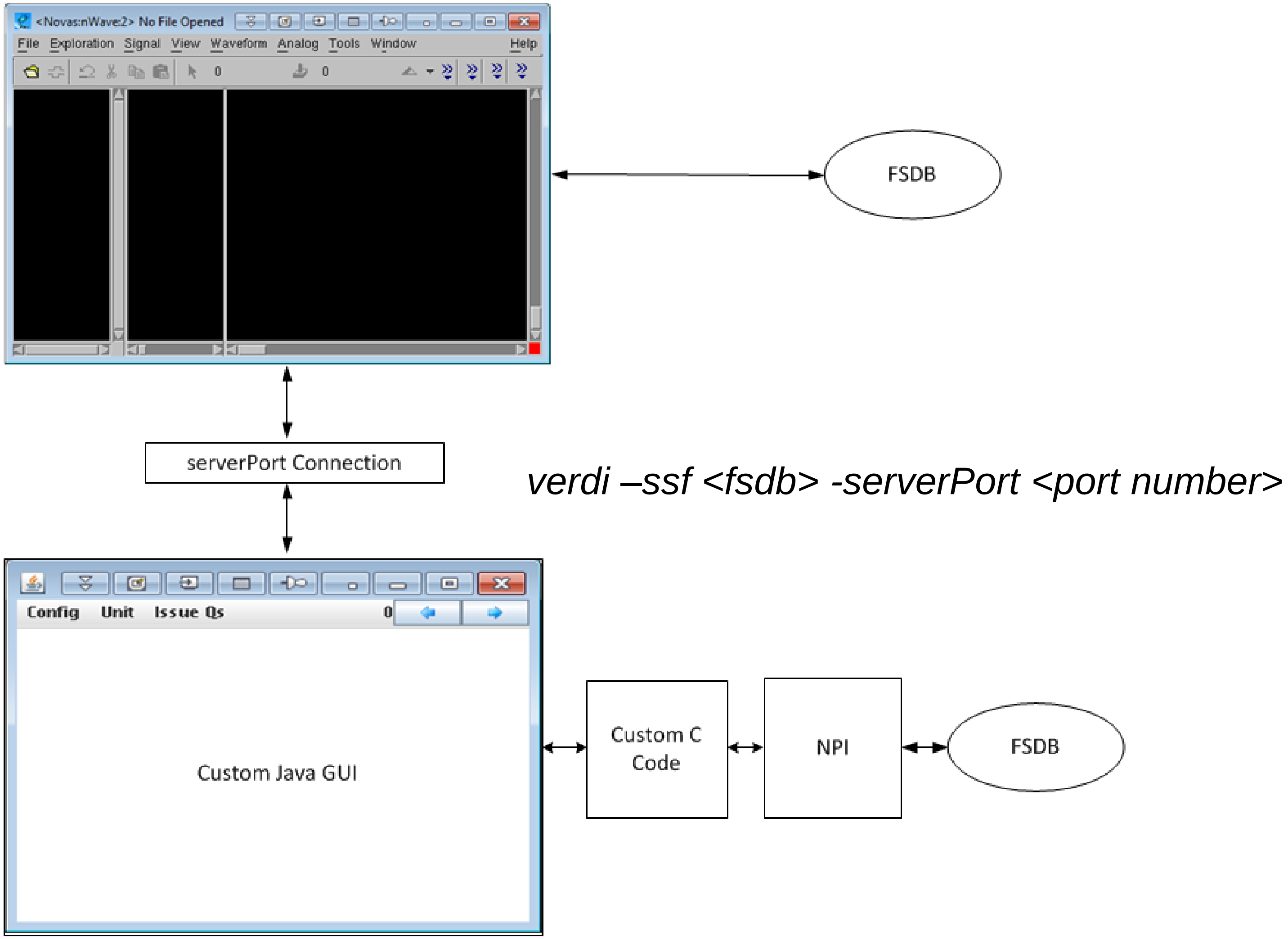
0	-	-
1	-	-
2	-	-
3	-	-
4	-	-
5	-	-
6	-	-

R2

R2 uop	go	srcA	srcB	srcC	srcD	srcP	dstX	dstY	dstP
- - -	-	- -	- -	- -	- -	- -	- -	- -	- -
- - -	-	- -	- -	- -	- -	- -	- -	- -	- -
- - -	-	- -	- -	- -	- -	- -	- -	- -	- -

Dynamic Register GUI

General Setup

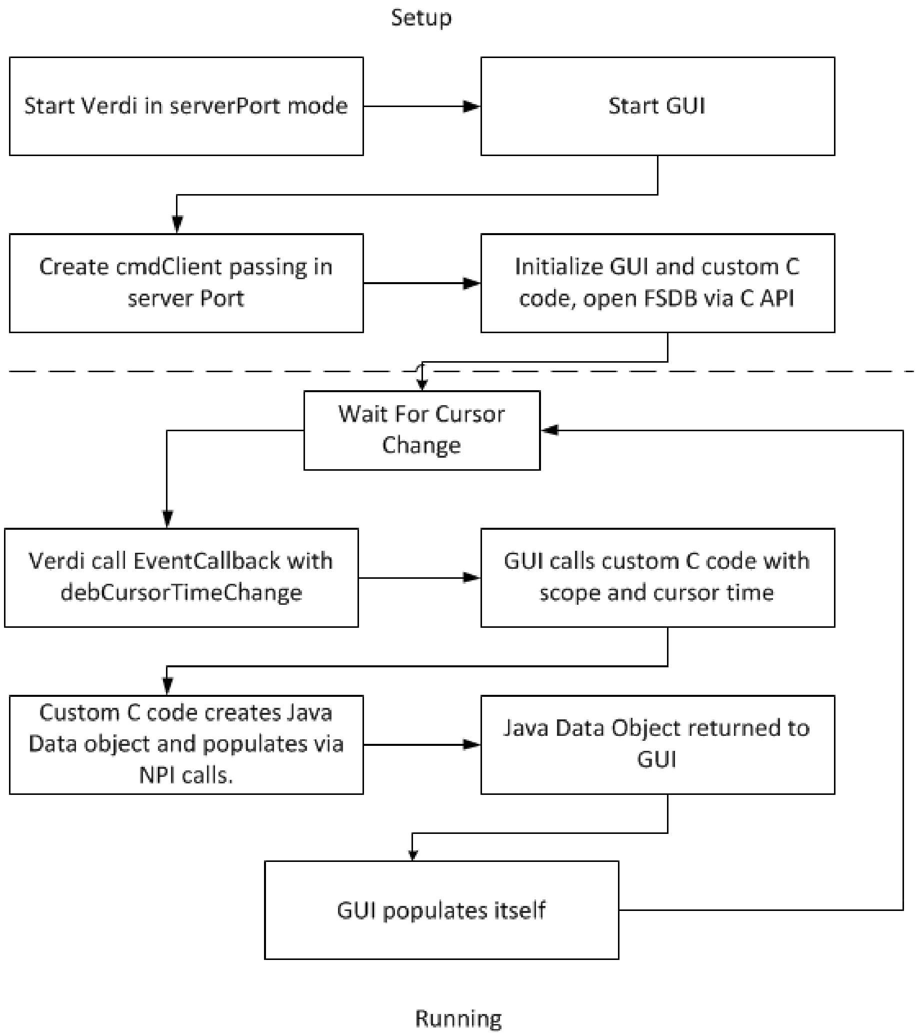


<Novas:nWave:2> No File Opened
File Exploration Signal View Waveform Analog Tools Window Help
0
0
FSDB
serverPort Connection
Config Unit Issue Qs 0
Custom Java GUI
Custom C Code
NPI
FSDB

verdi –ssf <fsdb> -serverPort <port number>

Dynamic Register GUI

Program Flow

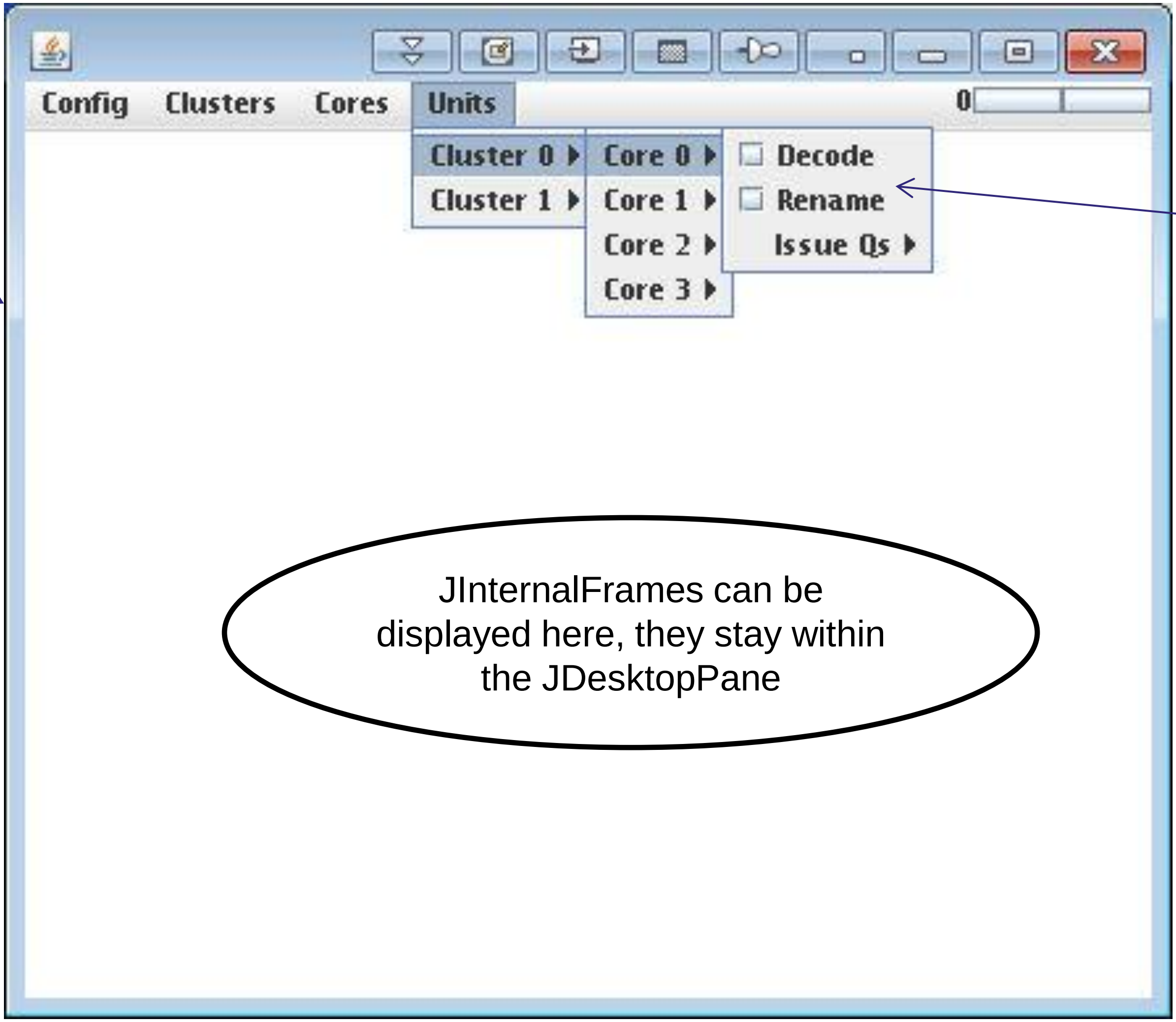


Setup
Start Verdi in serverPort mode
Start GUI
Create cmdClient passing in server Port
Initialize GUI and custom C code, open FSDB via C API
Wait For Cursor Change
Verdi call EventCallback with debCursorTimeChange
GUI calls custom C code with scope and cursor time
Custom C code creates Java Data object and populates via NPI calls.
Java Data Object returned to GUI
GUI populates itself
Running

Dynamic Register GUI

Top Level

JDesktopPane

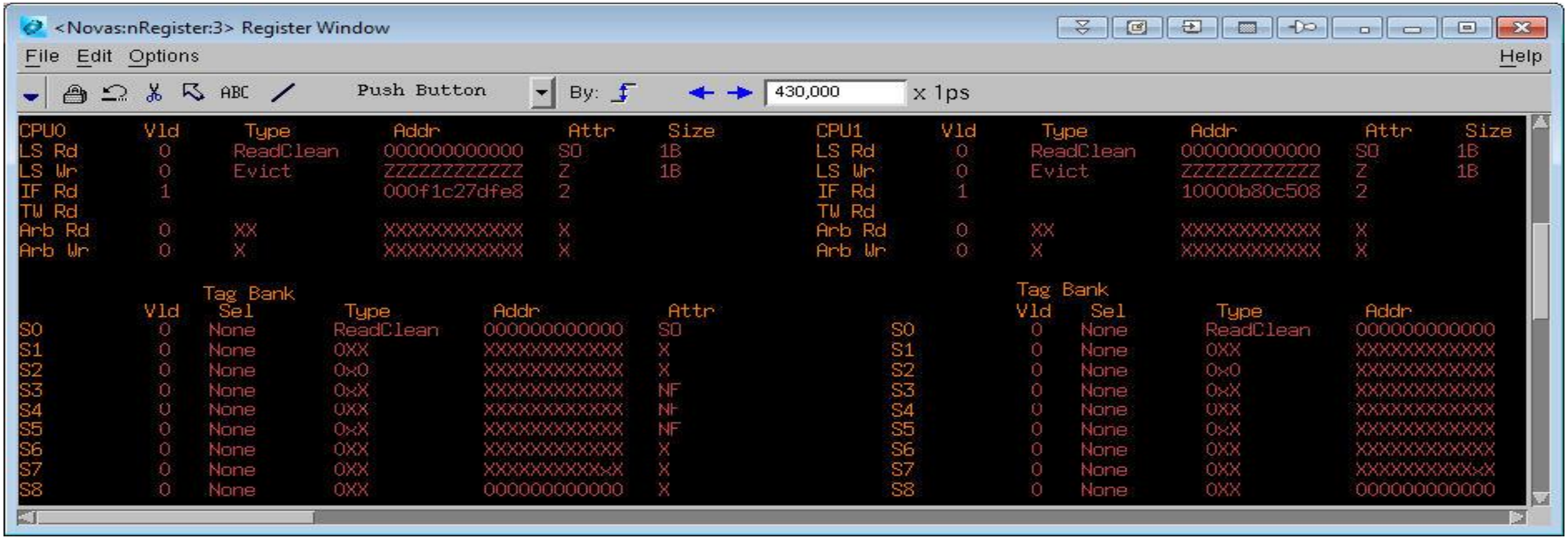


Opens JFrame containing unit information

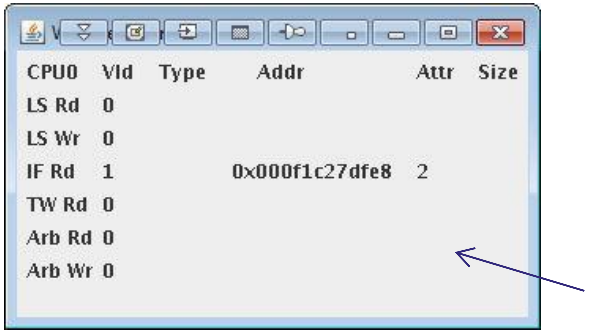
JInternalFrames can be displayed here, they stay within the JDesktopPane

Dynamic Register GUI

Vs Register Window Looks



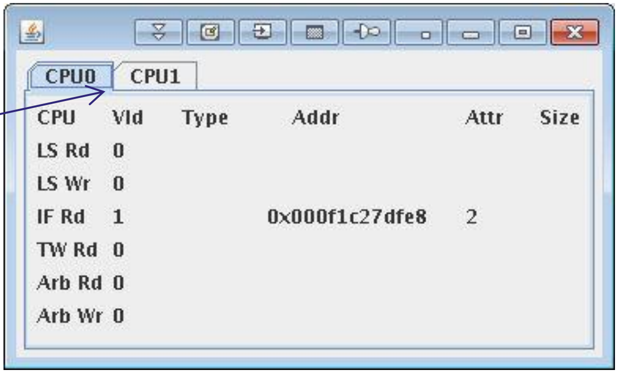
```
CPU0      Vld    Type        Addr          Attr   Size        CPU1      Vld    Type        Addr          Attr   Size
LS Rd      0     ReadClean   000000000000  S0     1B          LS Rd      0     ReadClean   000000000000  S0     1B
LS Wr      0     Evict       ZZZZZZZZZZZZ  Z      1B          LS Wr      0     Evict       ZZZZZZZZZZZZ  Z      1B
IF Rd      1                 000f1c27dfe8  2                  IF Rd      1                 10000b80c508  2
TW Rd                                                         TW Rd
Arb Rd     0     XX          XXXXXXXXXXXX  X                  Arb Rd     0     XX          XXXXXXXXXXXX  X
Arb Wr     0     X           XXXXXXXXXXXX  X                  Arb Wr     0     X           XXXXXXXXXXXX  X

               Tag Bank                                                            Tag Bank
          Vld   Sel        Type        Addr           Attr                         Vld     Sel       Type        Addr
S0         0    None       ReadClean   000000000000   S0                  S0        0      None      ReadClean   000000000000
S1         0    None       0XX         XXXXXXXXXXXX   X                   S1        0      None      0XX         XXXXXXXXXXXX
S2         0    None       0x0         XXXXXXXXXXXX   X                   S2        0      None      0x0         XXXXXXXXXXXX
S3         0    None       0xX         XXXXXXXXXXXX   NF                  S3        0      None      0xX         XXXXXXXXXXXX
S4         0    None       0XX         XXXXXXXXXXXX   NF                  S4        0      None      0XX         XXXXXXXXXXXX
S5         0    None       0xX         XXXXXXXXXXXX   NF                  S5        0      None      0xX         XXXXXXXXXXXX
S6         0    None       0XX         XXXXXXXXXXXX   X                   S6        0      None      0XX         XXXXXXXXXXXX
S7         0    None       0XX         XXXXXXXXXXxX   X                   S7        0      None      0XX         XXXXXXXXXXxX
S8         0    None       0XX         000000000000   X                   S8        0      None      0XX         000000000000
```



```
CPU0    Vld   Type     Addr                Attr   Size
LS Rd   0
LS Wr   0
IF Rd   1              0x000f1c27dfe8      2
TW Rd   0
Arb Rd  0
Arb Wr  0
```

CPUs are in a tabbedPane

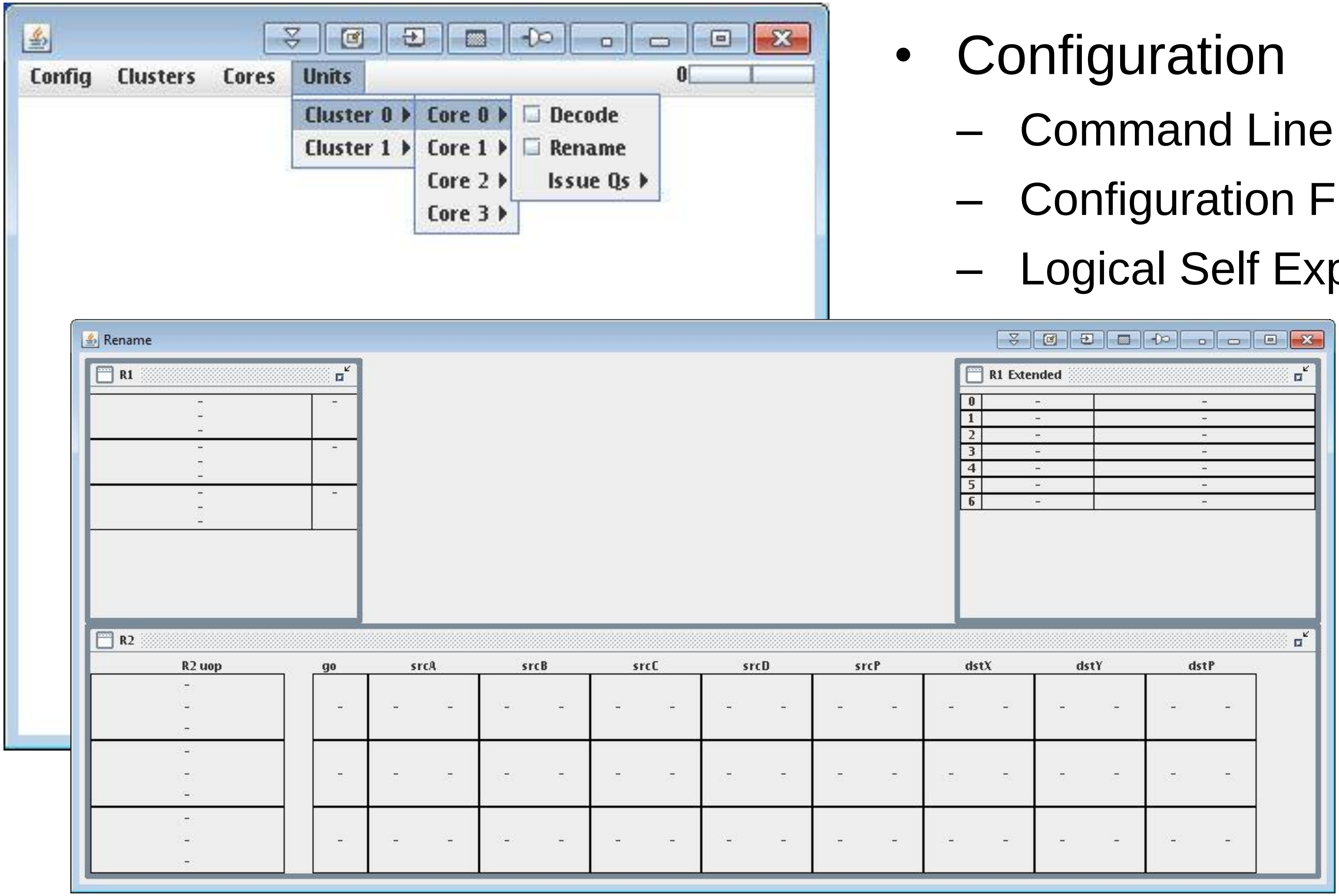
Beauty takes time



```
CPU     Vld   Type     Addr                Attr   Size
LS Rd   0
LS Wr   0
IF Rd   1              0x000f1c27dfe8      2
TW Rd   0
Arb Rd  0
Arb Wr  0
```

Dynamic Register GUI

Configuration



R1

- - -	-
- - -	-
- - -	-

R1 Extended

0	-	-
1	-	-
2	-	-
3	-	-
4	-	-
5	-	-
6	-	-

R2

R2 uop	go	srcA	srcB	srcC	srcD	srcP	dstX	dstY	dstP
- - -	-	- -	- -	- -	- -	- -	- -	- -	- -
- - -	-	- -	- -	- -	- -	- -	- -	- -	- -
- - -	-	- -	- -	- -	- -	- -	- -	- -	- -

- Configuration
  - Command Line
  - Configuration Files
  - Logical Self Exploration

Conclusions

Register Window

- Pros
  - Simple To Use
  - Single System for Debug with a wave dump
- Cons
  - Poor reuse
  - Limited graphical options
  - File management issues for large projects

Conclusions

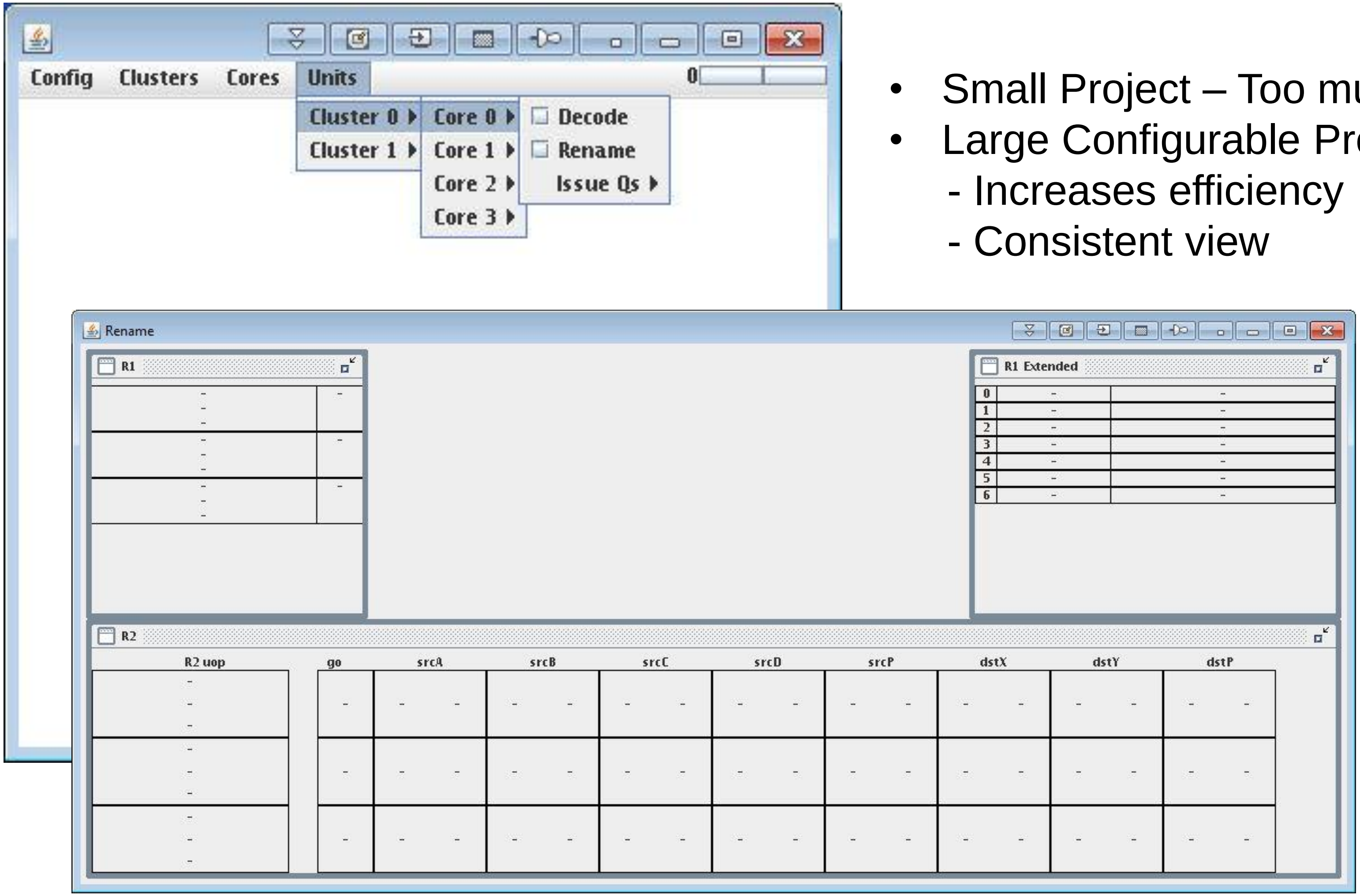
Dynamic Register GUI

- Pros
  - Reuse
  - GUI population from multiple sources
  - Dynamically configurable
  - Limited only by engineer’s imagination and time

- Cons
  - Complex
  - Most RTL designers/engineers have limited GUI Experience
  - Multiple languages
  - Multiple places need to be edited

Dynamic Register GUI

Conclusion – Is it worth the work?



R1

- - -	-
- - -	-
- - -	-

R1 Extended

0	-	-
1	-	-
2	-	-
3	-	-
4	-	-
5	-	-
6	-	-

R2

R2 uop	go	srcA	srcB	srcC	srcD	srcP	dstX	dstY	dstP
- - -	-	- -	- -	- -	- -	- -	- -	- -	- -
- - -	-	- -	- -	- -	- -	- -	- -	- -	- -
- - -	-	- -	- -	- -	- -	- -	- -	- -	- -

- Small Project – Too much work
- Large Configurable Project
  - Increases efficiency
  - Consistent view

Questions?

Thank You