SPADIC 1.1 status

Michael Krieger

CBM Collaboration Meeting

16.09.2015

Changes SPADIC 1.0 →1.1

- As a black box: compatible to 1.0
- Fix amplifier instability
- Fix serializer glitch
- Fix comparator bug (cannot use negative thresholds, rising edge missing, etc.) → *already done since February*
- Use updated CBMnet core → *done*

Needs reimplementation of the digital part (semi-custom design flow)

Progress

in April

"in progress, submit in May"

- $\frac{\text{available time}}{\text{required time}}$ *slightly* overestimated
- amplifier not yet fixed (now high priority by P. Fischer)
- difficulties of getting the implementation scripts running

Digital implementation

- scripts were easy to break and hard to maintain (they have been copy/pasted from several sources over a few generations)
- complete overhaul to fix inconsistencies and make debugging problems feasable
- in April: $\approx$ 500 lines of code changed
- now: $\approx$ 2000 lines of code changed
- had to revert to older version of Cadence tools to get them running

Example: pin positions (old)

```
set ypos [expr $dieSizeY]
editPin -side Right -use CLOCK -layer $pinLayerRight -unit MICRON -pinWidth $pinW \
  -assign $xPosRight $ypos -pin clk10 -snap USERGRID -fixedPin 1
addCustomText M3 "clk10" $xPosRight $ypos $labelH

set pinNo 2
set ypos [expr $dieSizeY - $offsetRightToPin0 - ($pinNo * $padWidth)]
editPin -side Right -use CLOCK -layer $pinLayerRight -unit MICRON -pinWidth $pinW \
  -assign $xPosRight $ypos -pin clk2 -snap USERGRID -fixedPin 1
addCustomText M3 "clk2" $xPosRight $ypos $labelH

set pinNo 5
set ypos [expr $dieSizeY - $offsetRightToPin0 - ($pinNo * $padWidth)]
editPin -side Right -use CLOCK -layer $pinLayerRight -unit MICRON -pinWidth $pinW \
  -assign $xPosRight $ypos -pin clk1 -snap USERGRID -fixedPin 1
addCustomText M3 "clk1" $xPosRight $ypos $labelH

set pinNo 12
set ypos [expr $dieSizeY - $offsetRightToPin0 - ($pinNo * $padWidth)]
editPin -side Right -use SIGNAL -layer $pinLayerRight -unit MICRON -pinWidth $pinW \
  -assign $xPosRight $ypos -pin resN -snap USERGRID -fixedPin 1
addCustomText M3 "resN" $xPosRight $ypos $labelH

set pinNo 15
set ypos [expr $dieSizeY - $offsetRightToPin0 - ($pinNo * $padWidth)]
editPin -side Right -use SIGNAL -layer $pinLayerRight -unit MICRON -pinWidth $pinW \
  -assign $xPosRight $ypos -pin dataIn -snap USERGRID -fixedPin 1
addCustomText M3 "dataIn" $xPosRight $ypos $labelH

set pinNo 16
set ypos [expr $dieSizeY - $offsetRightToPin0 - ($pinNo * $padWidth)]
editPin -side Right -use SIGNAL -layer $pinLayerRight -unit MICRON -pinWidth $pinW \
  -assign $xPosRight $ypos -pin triggerOut -snap USERGRID -fixedPin 1
addCustomText M3 "triggerOut" $xPosRight $ypos $labelH

set pinNo 20
set ypos [expr $dieSizeY - $offsetRightToPin0 - ($pinNo * $padWidth)]
editPin -side Right -use SIGNAL -layer $pinLayerRight -unit MICRON -pinWidth $pinW \
  -assign $xPosRight $ypos -pin enableReadout -snap USERGRID -fixedPin 1
addCustomText M3 "enableReadout" $xPosRight $ypos $labelH

set pinNo 21
set ypos [expr $dieSizeY - $offsetRightToPin0 - ($pinNo * $padWidth)]
editPin -side Right -use SIGNAL -layer $pinLayerRight -unit MICRON -pinWidth $pinW \
  -assign $xPosRight $ypos -pin readoutEnabled -snap USERGRID -fixedPin 1
addCustomText M3 "readoutEnabled" $xPosRight $ypos $labelH

set pinNo 27
set ypos [expr $dieSizeY - $offsetRightToPin0 - ($pinNo * $padWidth)]
editPin -side Right -use SIGNAL -layer $pinLayerRight -unit MICRON -pinWidth $pinW \
  -assign $xPosRight $ypos -pin linkActive -snap USERGRID -fixedPin 1
addCustomText M3 "linkActive" $xPosRight $ypos $labelH
```

- command for placing a pin including all options repeated $n \approx 200$ times
- is it *really* n times the same command? if not, is it on purpose? is there a copy/paste error?

Example: pin positions (new)

```
# Definition of the top-level IO pins on the top, right, and bottom edges
# of the chip.
variable ioPins {{
    top {2 left {
        {miscOutTop {0 4 10}}
        {miscInTop {15 18 20 24}}
    }}
    right {-2.5 down {
        {clk10 0 CLOCK 2.5}
        {clk2 2 CLOCK}
        {clk1 5 CLOCK}
        {resN 12}
        {dataIn 15}
        {triggerOut 16}
        {enableReadout 20}
        {readoutEnabled 21}
        {linkActive 27}
        {SDA_out 28 {} -40}
        {SDA_in 28}
        {SCL 29}
        {resN1 32}
        {resN2 33}
        {resN10 34}
        {serdesReady 35}
        {userpin1 37}
        {userpin2 38}
        {dataOutA 45}
        {dataOutB 47}
    }}
    bottom {2 left {
        {miscOutBottom {0 4 10 14}}
        {miscInBottom {18 20 24}}
    }}
} 95}

# Definition of the pins for the connections to the analog part that follow
# no particular pattern.
variable analogPins {{left 58.5 down} {
    {readbackShiftReg 0}
    {serinShiftReg 1.5}
    {ck1ShiftReg 5}
    {ck2ShiftReg 11}
    {loadShiftReg 17}
    {seroutShiftReg 4464}
    {intAnalogTrigger 4470.5}
} 1}

# Definition of the ADC output pins from each channel. They are placed in
# two groups starting from the top and bottom edges of a channel moving
# inwards.
variable adcOutputPins {{left 312.5 down} {
    {HN3LN2 LN3HN2 LN1HN0 HN1LN0} down
    {HN5LN4 LN5HN4 LN7HN6 HN7LN6} up
} 3.5 1}
```

- define pin names and positions in one place
- define the necessary command with options separately
- place the pins in a few simple steps:

```
foreach side {top right bottom} {
    placeIoPins $dieBox $side
}

placeAnalogPins $dieBox

foreach channel [::spadic::allChannels] {
    placeAdcOutputPins $dieBox $channel
}
foreach group [::spadic::analogGroups] {
    setAdcClockPins $dieBox $group
}
placeAdcDecouplingPins $dieBox
```

Example: RAM placement (old)

RAMs (>40 in total) manually shifted to their positions

```
#place ECA relative to 7A
relativePlace \
  susTop/groupA/channelGroup/epochChannel/outputBuffer/fifoRamDual16/sramFaradayDual16 \
  susTop/groupA/channelGroup/channelLoop[7].channel/hitLogic/outputBuffer/fifoRamDual16/sramFaradayDual16 \
  -relation R -alignedBy T \
  -xOffset [expr $xChannelWidth - $xRamD16 - $xRamGap + (($xSpaceSpadicLogic - $xRamD16)/2.0)]
set bb [getObjFPlanBoxList Instance \
  susTop/groupA/channelGroup/epochChannel/outputBuffer/fifoRamDual16/sramFaradayDual16]
set bb [list [lindex $bb 0] [expr [lindex $bb 1] - $yRamRouteBlk] [lindex $bb 2] [expr [lindex $bb 3] + $yRamRouteBlk]]
createRouteBlk -layer M1 -box $bb

#place OFA relative to 11A
relativePlace \
  susTop/groupA/orderingFifo/fifoRamDual17/sramFaradayDual17 \
  susTop/groupA/channelGroup/channelLoop[11].channel/hitLogic/outputBuffer/fifoRamDual16/sramFaradayDual16 \
  -relation R -alignedBy T \
  -xOffset [expr $xChannelWidth - $xRamD16 - $xRamGap + (($xSpaceSpadicLogic - $xRamD17)/2.0)]
set bb [getObjFPlanBoxList Instance \
  susTop/groupA/orderingFifo/fifoRamDual17/sramFaradayDual17]
set bb [list [lindex $bb 0] [expr [lindex $bb 1] - $yRamRouteBlk] [lindex $bb 2] [expr [lindex $bb 3] + $yRamRouteBlk]]
createRouteBlk -layer M1 -box $bb

#place OBA relative to 15A
relativePlace \
  susTop/groupA/switch/fifoRamDual16/sramFaradayDual16 \
  susTop/groupA/channelGroup/channelLoop[15].channel/hitLogic/outputBuffer/fifoRamDual16/sramFaradayDual16 \
  -relation R -alignedBy T \
  -xOffset [expr $xChannelWidth - $xRamD16 - $xRamGap + (($xSpaceSpadicLogic - $xRamD16)/2.0)]
set bb [getObjFPlanBoxList Instance \
  susTop/groupA/switch/fifoRamDual16/sramFaradayDual16]
set bb [list [lindex $bb 0] [expr [lindex $bb 1] - $yRamRouteBlk] [lindex $bb 2] [expr [lindex $bb 3] + $yRamRouteBlk]]
createRouteBlk -layer M1 -box $bb
```

Example: RAM placement (new)

```
set grid [createGrid]

# create guides for channels and CBMnet
foreach channel [::spadic::allChannels] {
    createChannelGuide $grid $channel
}
createCbmnetGuide $grid

# place all RAMs
foreach channel [::spadic::allChannels] {
    set ram [::spadic::instance channelBuffer {*}$channel]
    set center [channelPosition $grid $channel]
    placeInstanceCenter $ram $center
}
foreach ram [::spadic::instance globalRams] \
        center [spadicGlobalRamPositions $grid] {
    placeInstanceCenter $ram $center
}
foreach ram [::spadic::instance cbmnetRams] \
        center [cbmnetRamPositions $grid] {
    placeInstanceCenter $ram $center
}
```

- define an evenly spaced grid
- place the RAMs on the grid

Example: for each

```
#for.. for each constraint mode and for each delay corner
create_analysis_view -name view_slow_functional    -constraint_mode slow_functional    -delay_corner slow_corner
create_analysis_view -name view_typical_functional -constraint_mode typical_functional -delay_corner typical_corner
create_analysis_view -name view_fast_functional    -constraint_mode fast_functional    -delay_corner fast_corner
create_analysis_view -name view_slow_test          -constraint_mode slow_test          -delay_corner slow_corner
create_analysis_view -name view_typical_test       -constraint_mode typical_test       -delay_corner typical_corner
create_analysis_view -name view_fast_test          -constraint_mode fast_test          -delay_corner fast_corner
foreach mode {functional test} {
    foreach corner {slow typical fast} {
        create_analysis_view -name view_${corner}_${mode} -constraint_mode ${corner}_${mode} -delay_corner ${corner}_corner
    }
}
```

- the two dots represent the desperation of the original author
- *if only there was a "for each" loop in Tcl*

Example: another highlight

```
#synthesize the design with low effort
synthesize -to_generic -effort high
synthesize -to_generic -effort high -incremental
synthesize -to_mapped -effort high
synthesize -to_mapped -effort high -incremental
synthesize -to_placed -effort high
synthesize -to_placed -effort high -incremental
```

no comment (no pun intended)

Current status

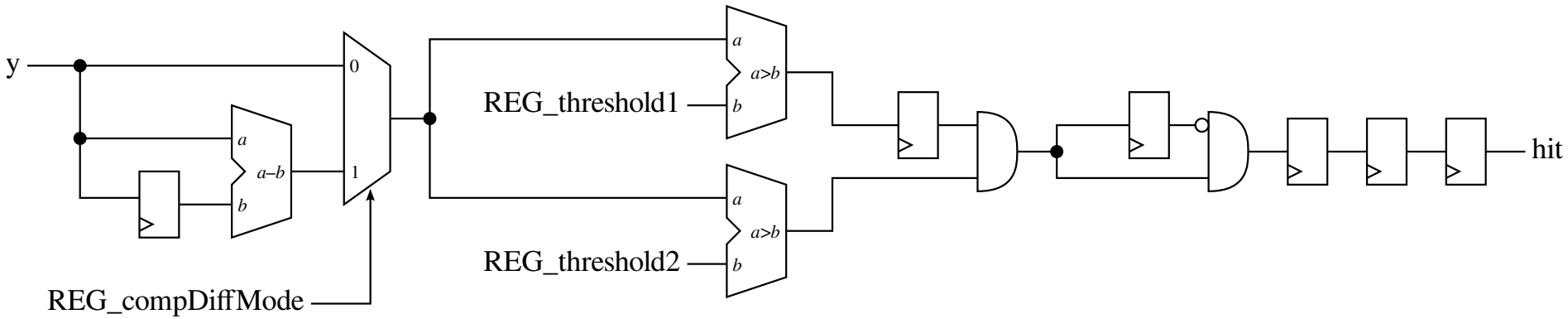
- can produce a layout file
- need to verify
  - design rules (no overlapping wires etc.)
  - correct timing
  - no serializer glitch
- P. Fischer will fix amplifier now
- submission possibly already next week! (maybe together with other CBM ASICs, currently in discussion)

Thank you for your attention.

New hit detector

	SPADIC 1.0 hit detector	SPADIC 1.1 hit detector
No. of clocks	2	1
Lines of code	280	70
No. of registers	45	14
No. of logic gates	≈ 240	≈ 140
Silicon area	$\approx 9000\,\mu m^2$	$\approx 4500\,\mu m^2$
Draw accurate picture	hard	easy

New hit detector



- 1 subtractor, 1 multiplexer, 2 comparators (9-bit)
- 2 AND-gates, 1 inverter
- 14 registers (9 + 1 + 1 + 3)
- delay of 3 clock cycles at output only for compatibility (can be removed in SPADIC 2.0, possibly also simplifying other logic)