*-> extended version with 6 output channels*

# WOTAN - low cost ultra small formfactor console for Magnetic Particle Imaging

*MPI control modul with 4 transmit channels (4x 250 kS/s, 8 bit, up to 4 MS/s possible) and one receive channel (2 MS/s, 12 bit).*

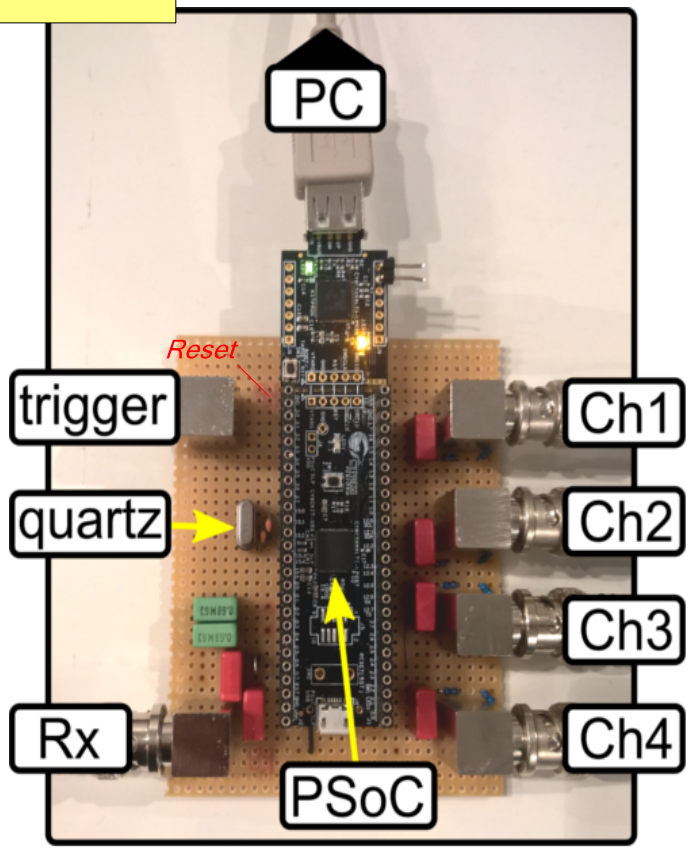
*Recording time: 15 ms. Data transfer over BLE (by adding the HC-05 bluetooth module or any bluetooth-UART module with 1.3 MBaud) with < 1.3 MBaud, or using the USBFS component included on the PSoC with 12.5 Mbps. Both interfaces allow writting new wave forms ("sequences") and receiving data.*

*Both interfaces appear as COM ports and can be used interchangeable.*

*Example scripts for python 3:*
*These scripts work with both interfaces. Since they appear as COM port, only the port number needs to be changed. Bluetooth is slower and might require a higher value for timeout.*

*1) "UI_WOTAN/generate_sequence.py": writting arbitrary wave forms to the module (example code for showing the API)*

*2) "UI_WOTAN/run_sequence.py": running the board via bluetooth or USBFS (fullspeed USB-to-UART).*




*External components (see following tabs for details):*

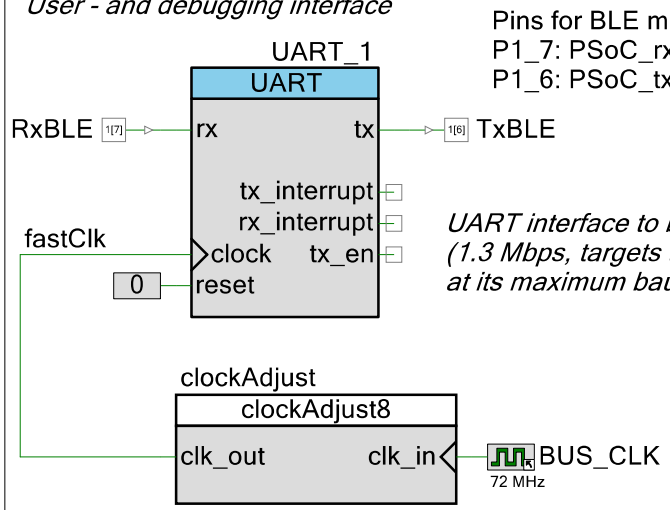
*CH1..Ch4: 4x 1uF capacitors for dc-blocking (GPIO 2.0, 2.3, 2.5 and 2.7 respectively).*
*4x 12 kOhm resistors to adjust the output voltage of the IDAC8_n components to 2V-pp (pull GPIO 2.0, 2.3, 2.5, 2.7 to GND with 12 kohm each).*

*Rx: 2x 1uF capacitors for the Rx-channel (dc-block, input_p=) and 1 Mohm resistor. 2x 680nF bypass capacitors for the two SAR_ADC.*

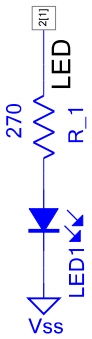
*XTAL: 8MHz quartz and 2x 33pF (GPIO 15.0, 15.1, GPIO 15.2 and 15.3 are used as GND for the oscillator)*

*User - and debugging interface*

Pins for BLE module:
P1_7: PSoC_rx -> BLE_tx
P1_6: PSoC_tx -> BLE_rx




*UART interface to bluetooth modem (1.3 Mbps, targets the HC-05 Module at its maximum baud rate)*

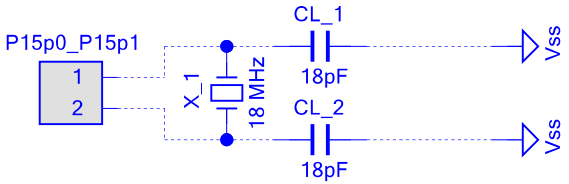



*clockAdjust settings:*

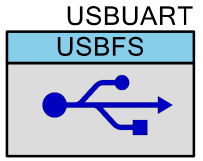
*ACCU_WORD =*
*= OVERSAMPLING * BAUDRATE * 2**ACCU_WIDTH / BUS_CLK*
*= 2576980*

*OVERSAMPLING = 8*
*BAUDRATE = 1382400*
*ACCU_WIDTH = 24*
*BUS_CLK = 72e6*

*Extern 6 MHz crytal (3...24 MHz can be used)*

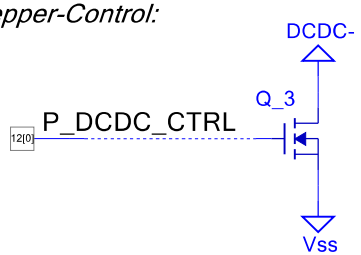



*USBUART for fast data transfer*
*(up to 12.5 Mbps, connected to micro USB Typ B)*

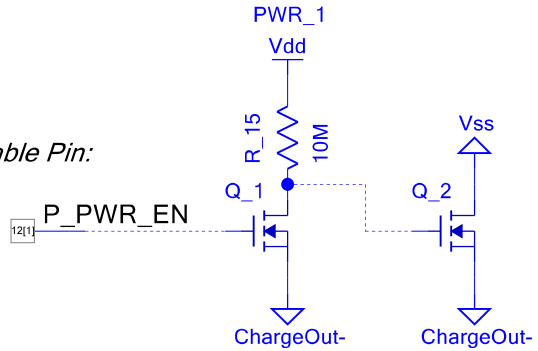


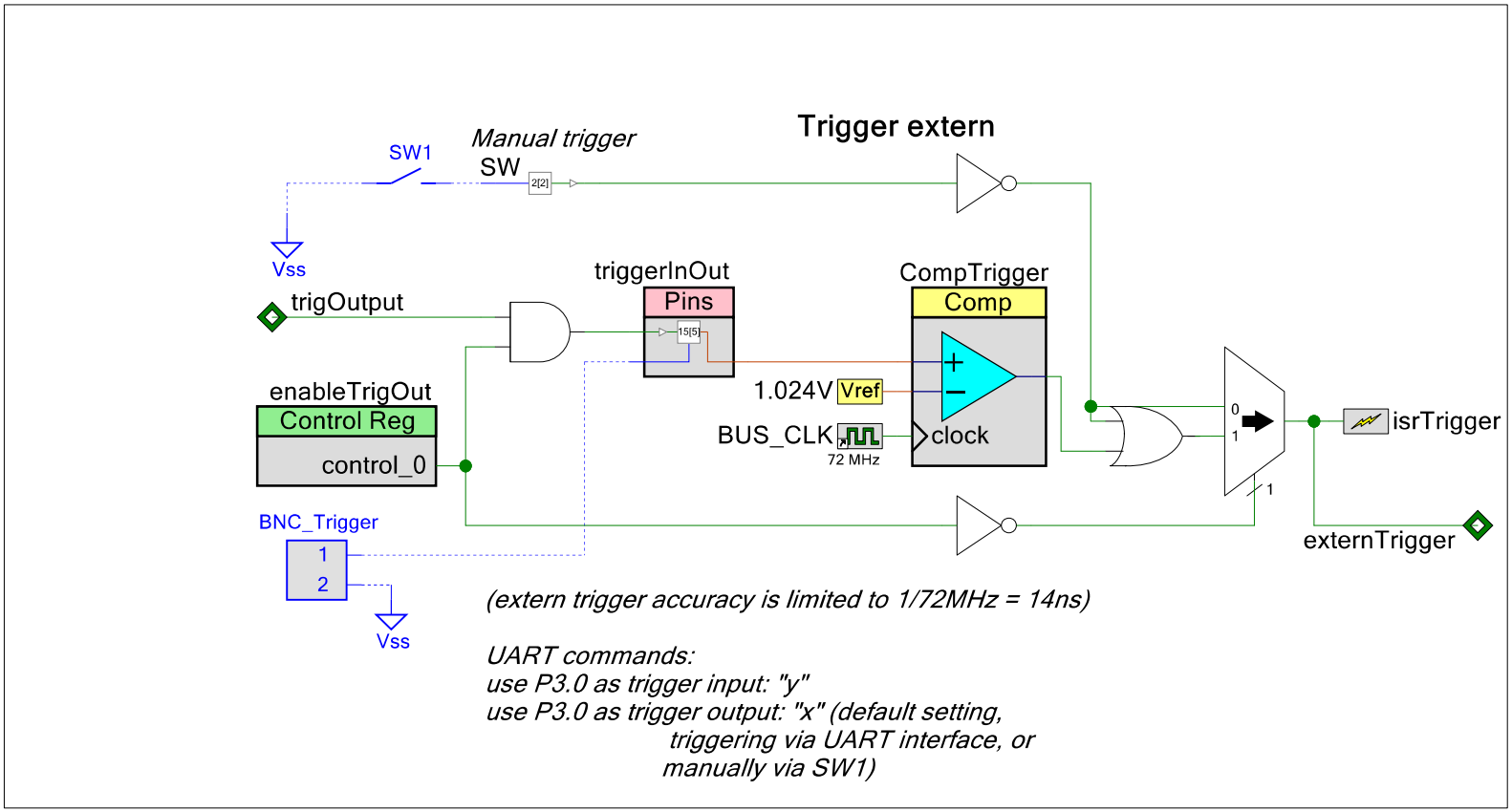

*Power Control:*

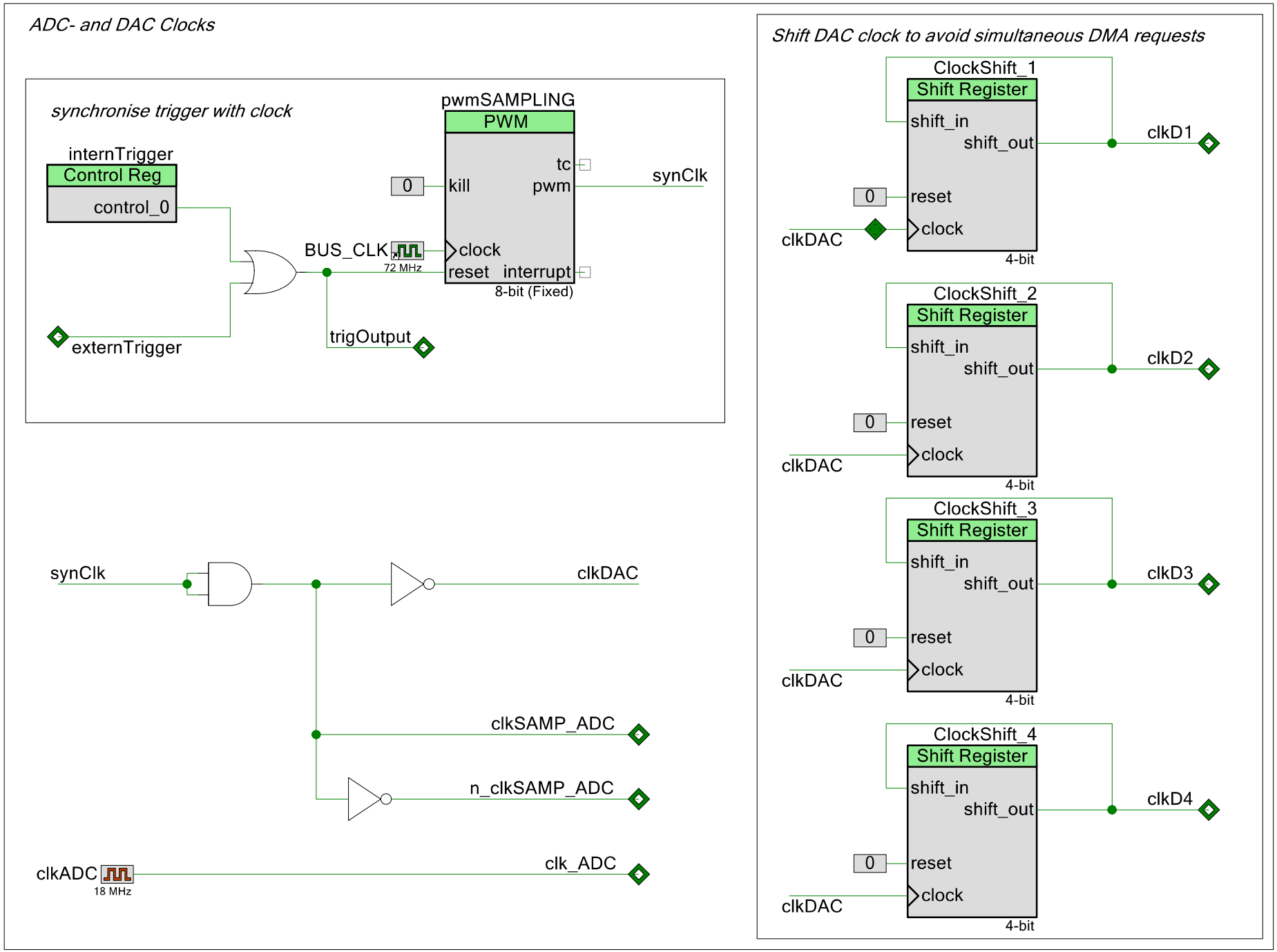
*Up-Stepper-Control:*



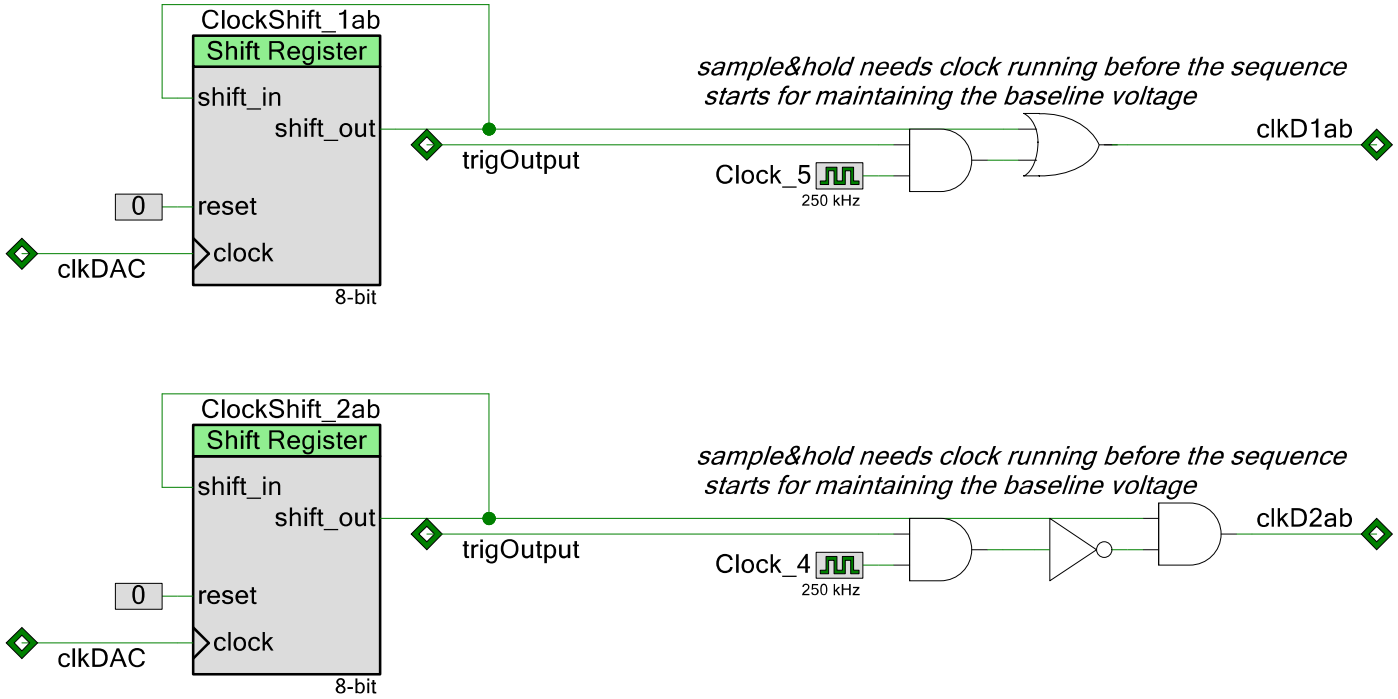

*Power Enable Pin:*

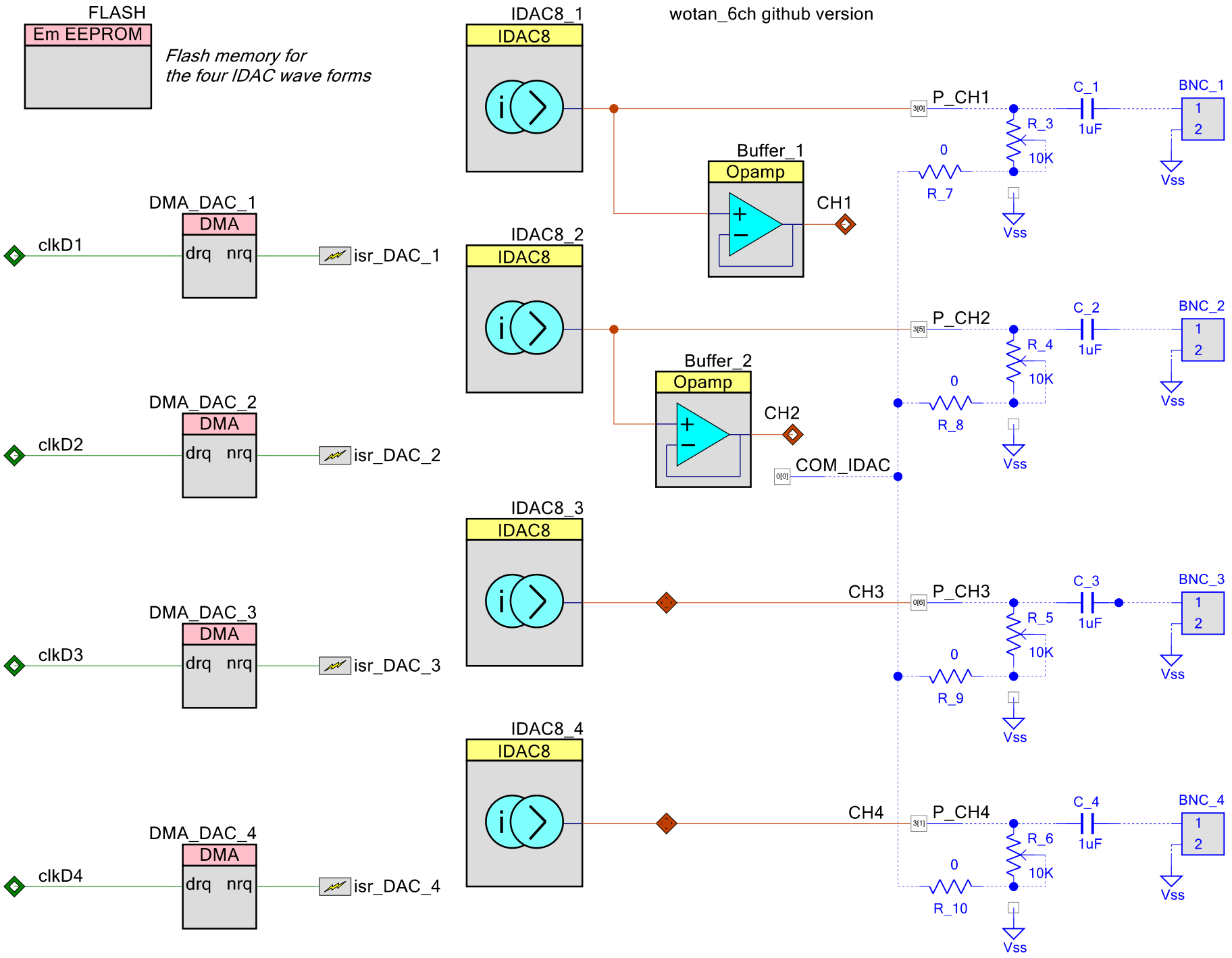


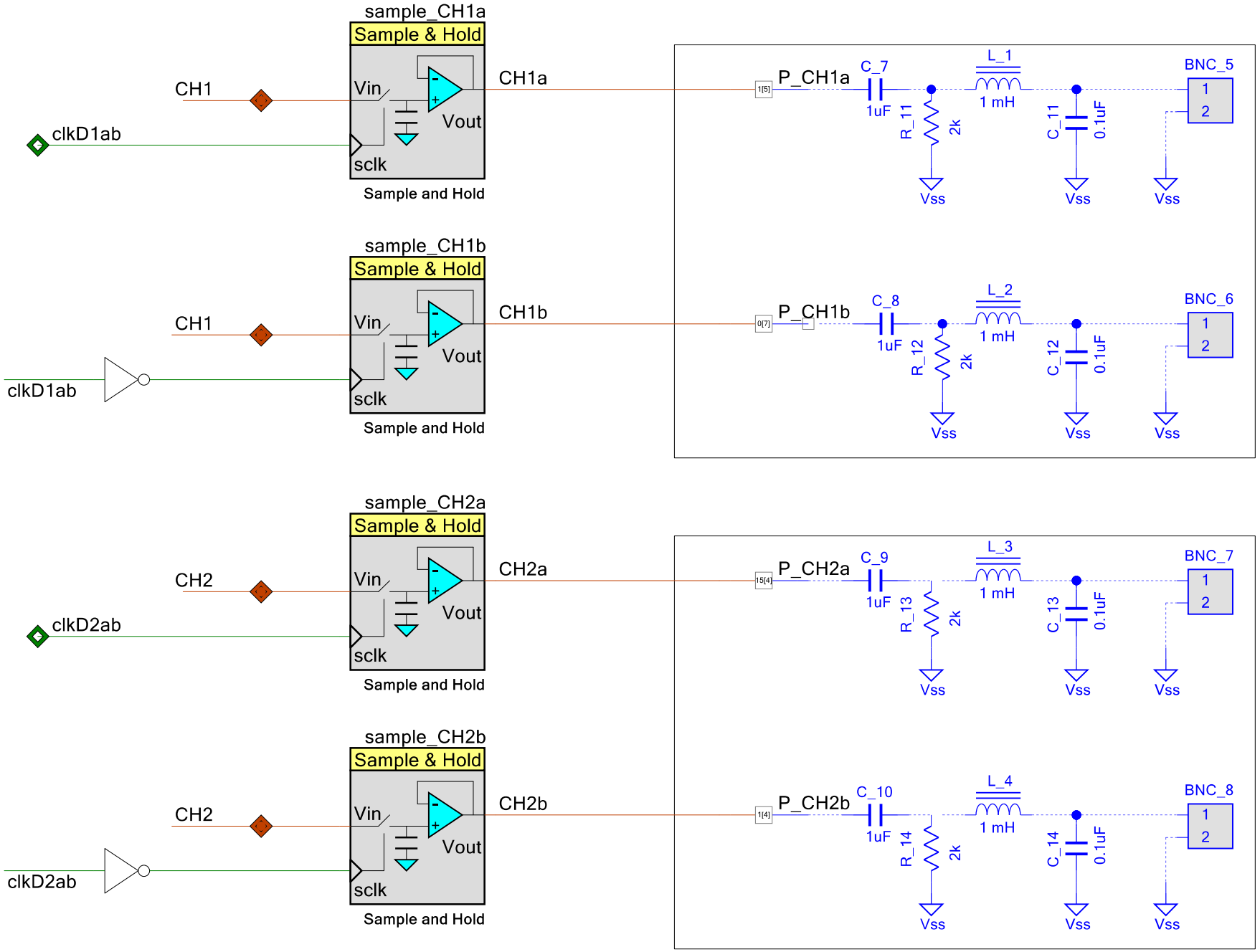

Trigger extern
Manual trigger
SW1
SW
2[2]
Vss
triggerInOut
Pins
15[5]
CompTrigger
Comp
trigOutput
enableTrigOut
Control Reg
control_0
1.024V
Vref
BUS_CLK
72 MHz
clock
0
1
1
isrTrigger
externTrigger
BNC_Trigger
1
2
Vss
(extern trigger accuracy is limited to 1/72MHz = 14ns)
UART commands:
use P3.0 as trigger input: "y"
use P3.0 as trigger output: "x" (default setting,
triggering via UART interface, or
manually via SW1)

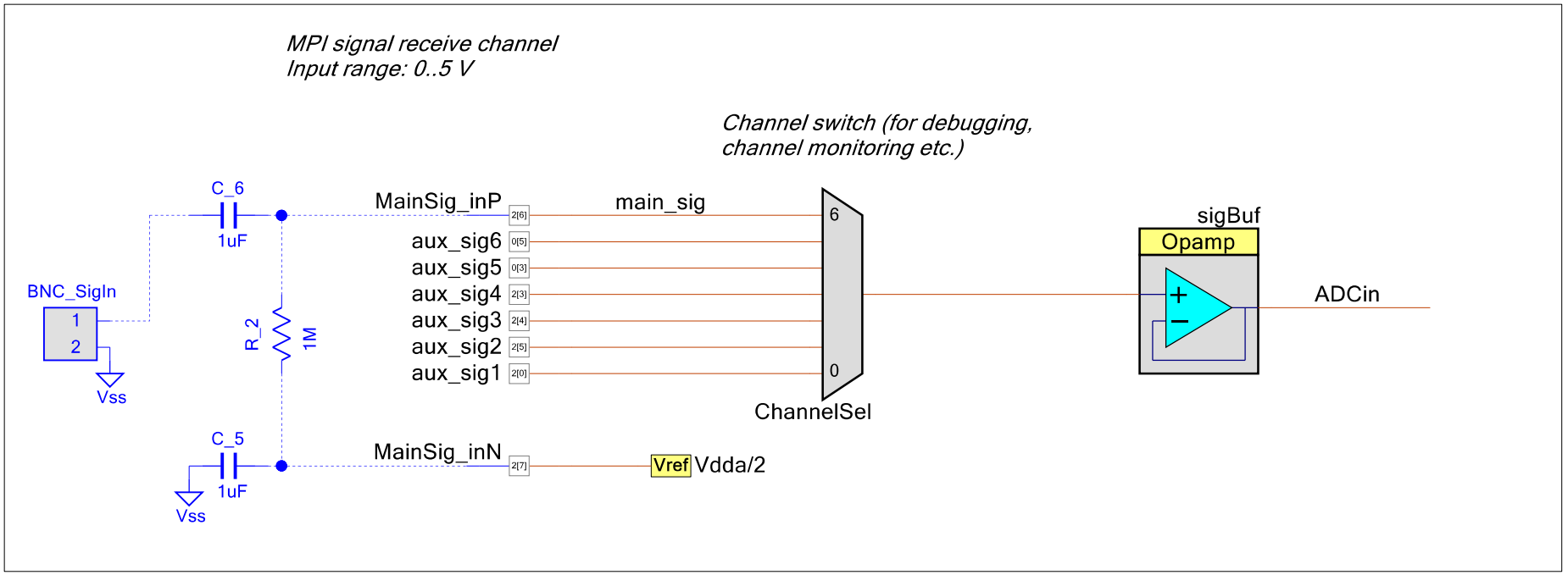

ADC- and DAC Clocks
synchronise trigger with clock
internTrigger
Control Reg
control_0
externTrigger
BUS_CLK
72 MHz
pwmSAMPLING
PWM
0
kill
tc
pwm
synClk
clock
reset
interrupt
8-bit (Fixed)
trigOutput
synClk
clkDAC
clkSAMP_ADC
n_clkSAMP_ADC
clkADC
18 MHz
clk_ADC
Shift DAC clock to avoid simultaneous DMA requests
ClockShift_1
Shift Register
shift_in
shift_out
clkD1
0
reset
clock
clkDAC
4-bit
ClockShift_2
clkD2
ClockShift_3
clkD3
ClockShift_4
clkD4

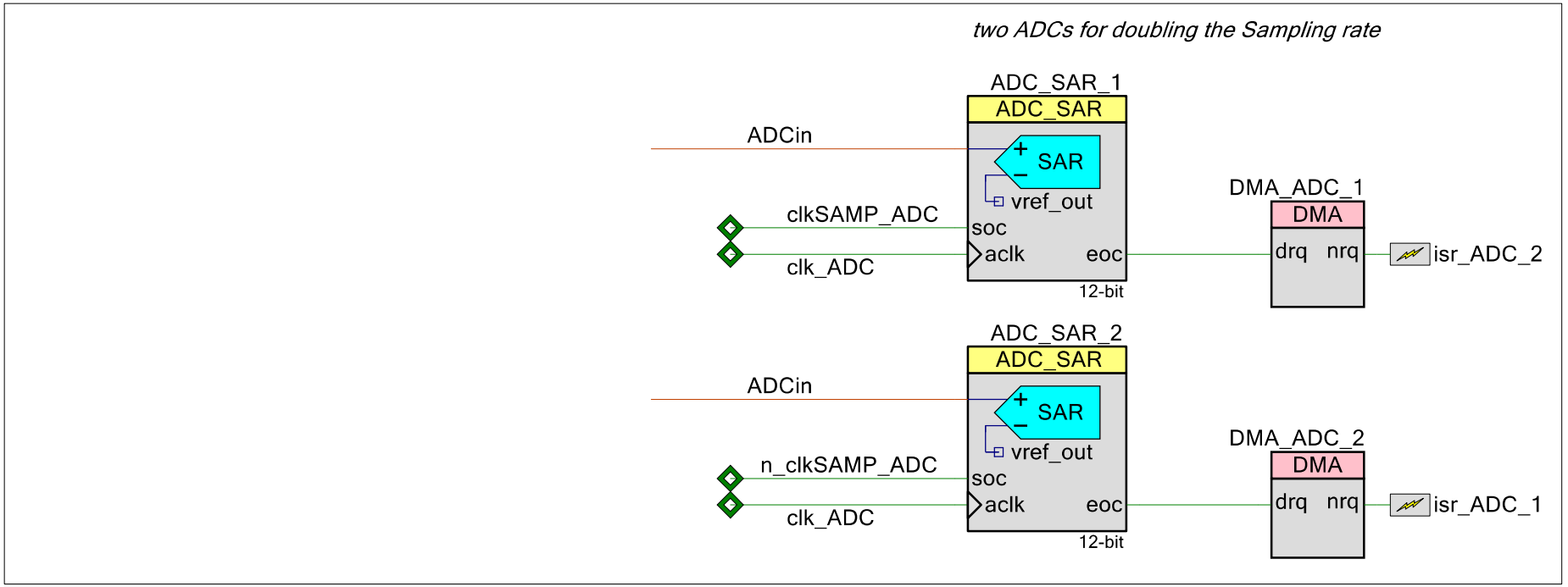
*Shift DAC clock for channel splitting (alternate samples between channels:*
*CH1 -> CH1a = even samples / CH1b = odd samples)*




FLASH
Em EEPROM
Flash memory for
the four IDAC wave forms
wotan_6ch github version
IDAC8_1
IDAC8
IDAC8_2
IDAC8
IDAC8_3
IDAC8
IDAC8_4
IDAC8
Buffer_1
Opamp
Buffer_2
Opamp
CH1
CH2
CH3
CH4
COM_IDAC
DMA_DAC_1
DMA
drq nrq
DMA_DAC_2
DMA
drq nrq
DMA_DAC_3
DMA
drq nrq
DMA_DAC_4
DMA
drq nrq
clkD1
clkD2
clkD3
clkD4
isr_DAC_1
isr_DAC_2
isr_DAC_3
isr_DAC_4
P_CH1
P_CH2
P_CH3
P_CH4
R_3
R_4
R_5
R_6
R_7
R_8
R_9
R_10
10K
0
C_1
C_2
C_3
C_4
1uF
BNC_1
BNC_2
BNC_3
BNC_4
Vss


sample_CH1a
Sample & Hold
CH1
clkD1ab
Vin
Vout
sclk
CH1a
Sample and Hold
sample_CH1b
Sample & Hold
CH1
clkD1ab
Vin
Vout
sclk
CH1b
Sample and Hold
sample_CH2a
Sample & Hold
CH2
clkD2ab
Vin
Vout
sclk
CH2a
Sample and Hold
sample_CH2b
Sample & Hold
CH2
clkD2ab
Vin
Vout
sclk
CH2b
Sample and Hold
P_CH1a
C_7
1uF
R_11
2k
L_1
1 mH
C_11
0.1uF
BNC_5
1
2
Vss
P_CH1b
C_8
1uF
R_12
2k
L_2
1 mH
C_12
0.1uF
BNC_6
P_CH2a
C_9
1uF
R_13
2k
L_3
1 mH
C_13
0.1uF
BNC_7
P_CH2b
C_10
1uF
R_14
2k
L_4
1 mH
C_14
0.1uF
BNC_8


MPI signal receive channel
Input range: 0..5 V
Channel switch (for debugging,
channel monitoring etc.)
C_6
1uF
BNC_SigIn
1
2
Vss
R_2
1M
C_5
1uF
Vss
MainSig_inP
MainSig_inN
main_sig
aux_sig6
aux_sig5
aux_sig4
aux_sig3
aux_sig2
aux_sig1
6
0
ChannelSel
sigBuf
Opamp
ADCin
Vref
Vdda/2


two ADCs for doubling the Sampling rate
ADC_SAR_1
ADC_SAR
ADCin
SAR
vref_out
clkSAMP_ADC
soc
clk_ADC
aclk
eoc
12-bit
DMA_ADC_1
DMA
drq
nrq
isr_ADC_2
ADC_SAR_2
ADC_SAR
ADCin
SAR
vref_out
n_clkSAMP_ADC
soc
clk_ADC
aclk
eoc
12-bit
DMA_ADC_2
DMA
drq
nrq
isr_ADC_1