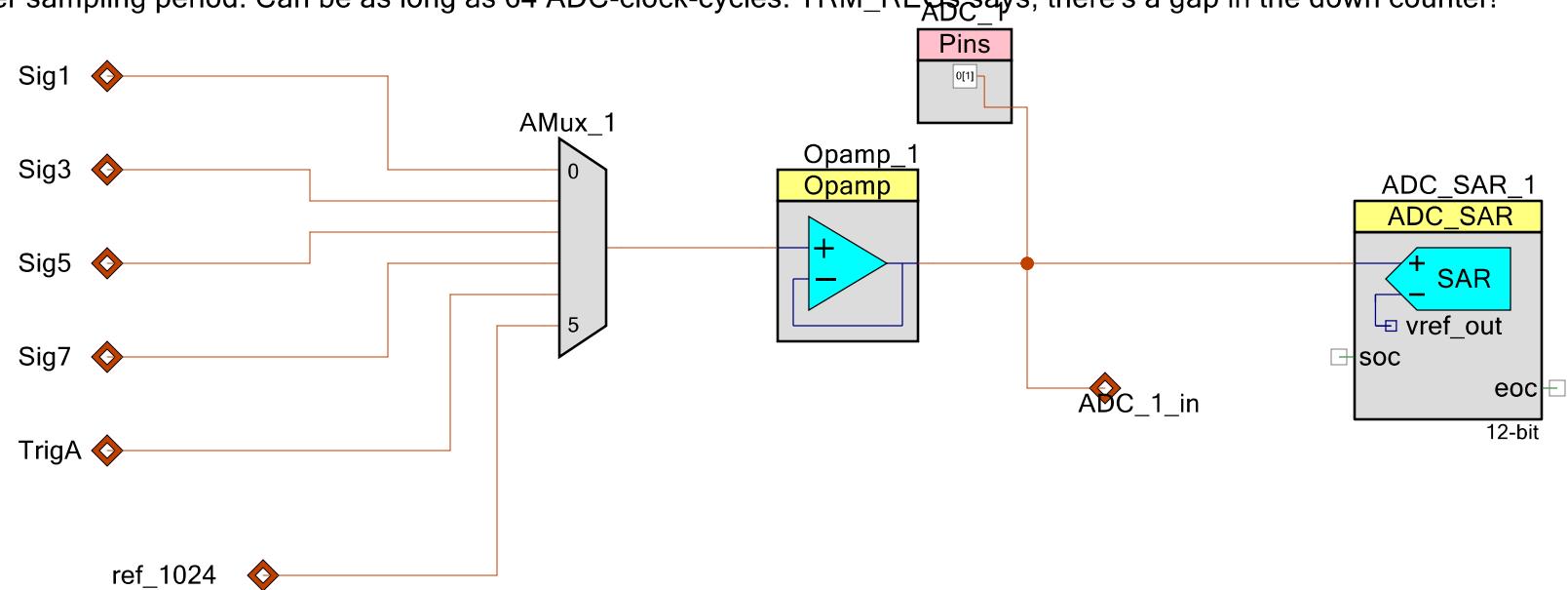


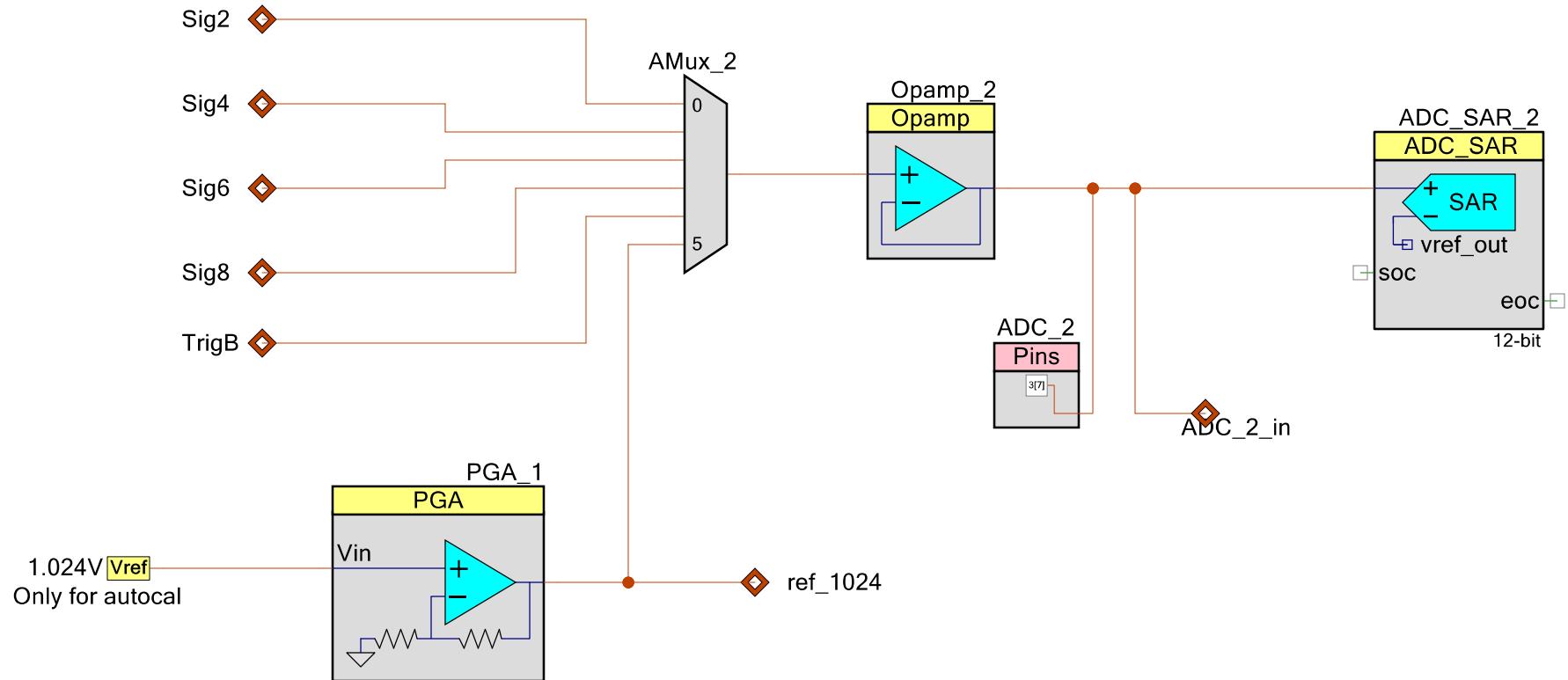
Approaches:

- 1: Using HW-controlled MUX: Doesn't work, because MUX must be GPIO-switches
- 2: DMA used for switching MUXes: Worth a try?
- 3: Delay using Control_Reg and Counter: Timing violation between digital and analog clock. Asked Cypress, no answer, yet.
- 4: Using eos-output of SAR-ADC to trigger interruptroutine for switching: May give highest speed but also highest CPU-load
- 5: Using continuous conversion and taking last value before switching MUX: No FIFO for results! Build own with DMA
- 6: SW-only version: One systick for MUX-switch, another for conversion: Will surely work, but is the slowest.
- 7: Using longer sampling period. Can be as long as 64 ADC-clock-cycles: TRM_REGS says, there's a gap in the down counter!



38.2.3 Input Sampling

The input sampling time can be programmed from the 1 to 64 cycles in register SARx_CSR[5:0] register bits. The user can also retain the earlier DAC value or clear it at the beginning of the new sampling clock. This is done in SARx_CSR0[3] register bit. The conversion time is 18 cycles for input sampling time up to four cycles. The maximum conversion time is 78 cycles for input sampling time of 64 cycles. The sampling time is chosen based on the source's input impedance so that the input settling time is lower than the sampling time



ADCs kalibrieren:

- AutoCal: Deckt nur PSoC-interne Fehler ab, lässt die Spannungsteiler unberücksichtigt
- ForcedCal: Von außen angelegte 0V und 10V nebst Cali-Aufruf per CAN deckt auch Spannungsteiler ab, lässt aber Betriebsspannungsschwankungen unberücksichtigt
- CombinedCal: Forced Cal nebst Messung der dabei anliegenden Vdda, zusätzlich AutoCal bezüglich Vdda MPM kommt ab Werk vorkalibriert

