JOHANNES KEPLER
UNIVERSITY LINZ

3-Bit Pipeline ADC

Design Project
Dominik Brandstetter (k12110413), Michael Köfinger (k01612781)
336.001 Advanced Integrated Circuit Design SS25
2025-06-02

JOHANNES KEPLER
UNIVERSITY LINZ
Altenberger Straße 69
4040 Linz, Austria
jku.at

Overview

- Introduction
- Specifications
- Principle of Operation
- System-Level Simulation
- Digital Core
- ADC Macro Model
- MDAC Design
- Circuit-Level Simulation

Introduction

- Tools
  - IIC_OSIC_TOOLS Docker container
  - IHP SG13G2 open-source PDK
  - Xschem/ngspice/verilator (analog/mixed-signal simulation)
  - Modelsim (digital simulation)
- Objective
  - Design and simulate a 3-bit pipeline ADC w/ 1.5 bit architecture
  - Mixed-signal/level simulation w/ Xschem/ngspice/verilator
  - Digital core using RTL and gate-level simulation
  - Design CMOS OTA
  - ADC: macro models
  - Switches: ideal

Specifications

- $n = 3\,\text{bit}$
- $n_{1,2} = 1.5\,\text{bit}$
- $n_3 = 1\,\text{bit}$
- $f_s = 2\,\text{MHz}$
- $V_{DD} = 1.5\,\text{V}$
- $V_{ref} = V_{DD}$

Principle of Operation

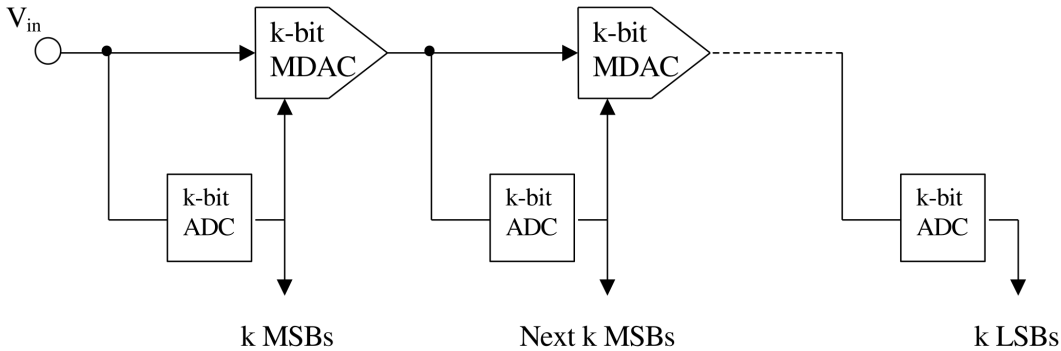

Figure 1: Block diagram.

System-Level Simulation

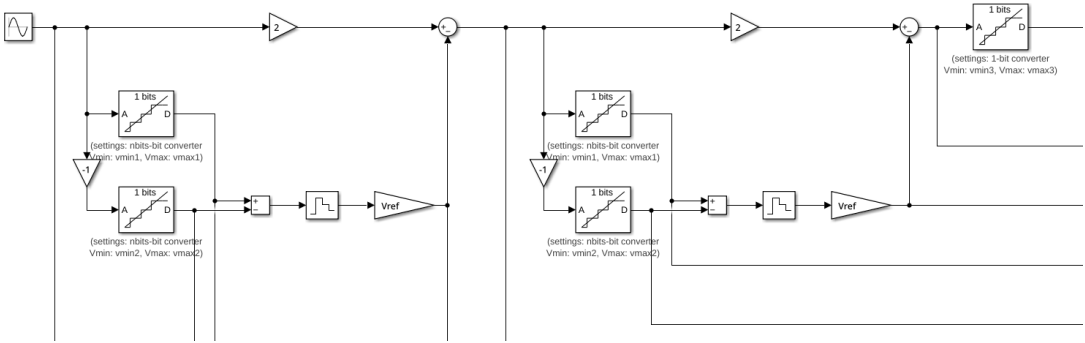

Figure 2: Simulink model.

System-Level Simulation cont'

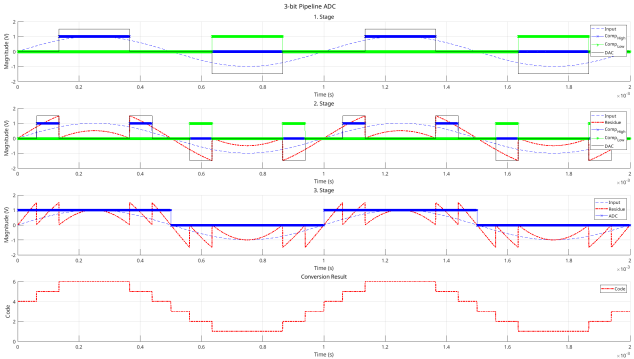

Figure 3: $A_{\mathrm{sig}} = 1\,\mathrm{V_{peak}}$ $\mathrm{f_{sig}} = 1\,\mathrm{kHz}$

3-bit Implementation w/ 1.5-bit Architecture

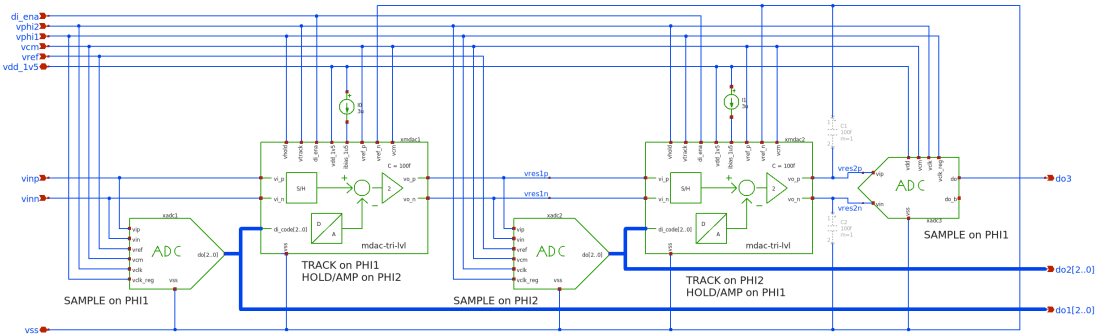

Figure 4: Schematic of ADC core.

Digital Core

- Add ADC outputs from each stage
- RTL simulation using Modelsim and Xschem
- Digital models generated from Ngspice circuits
- HDL (Verilog) used for digital model implementation
- Verilator creates a .so (shared object) file (referenced by Xschem symbol)

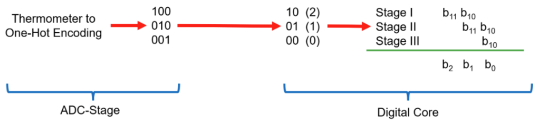

Figure 5: Summation.

Digital Core cont’

- Modelsim simulation

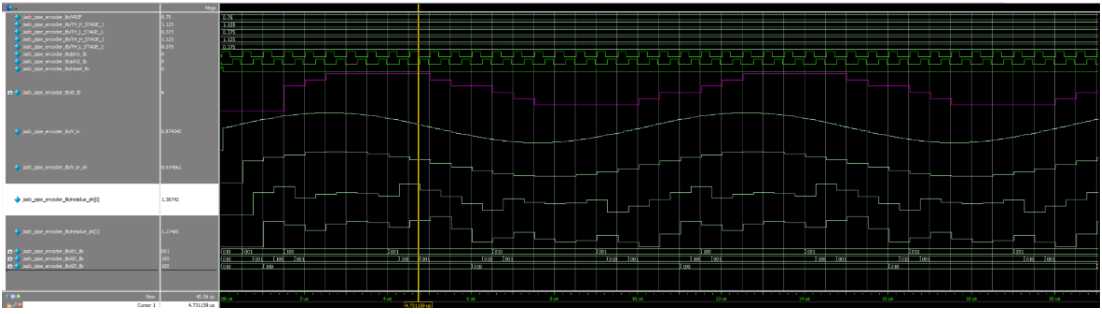

Figure 6: Summation.

Digital Core cont’

- Synthesis using Yosys (generating a xspice-file)
- Target size 100 x 100 µm

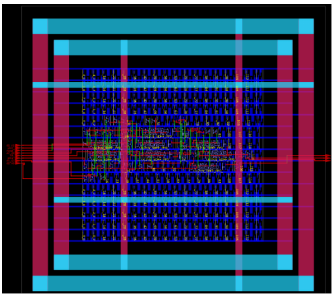

Figure 7: Layout after P&R.

Tri-Level ADC Macro Model

- Ideal comparator implemented using ADC bridges
- Data is latched during vclk and synchronized on the rising edge of vclk_reg
- Use DAC bridges to convert analog signals to digital format

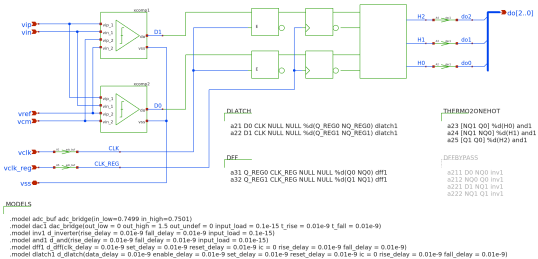

Figure 8: ADC macro model.

MDAC Design

- Sample/Track and hold, k-bit DAC, subtract and amplify

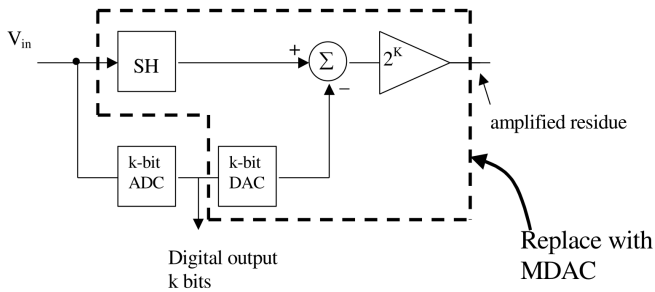

Figure 9: MDAC block diagram.

MDAC Design cont'

- $V_{\text{out}}[n+1] = -2V_{\text{in}}[n] - V_{\text{ref}}[n+1]$

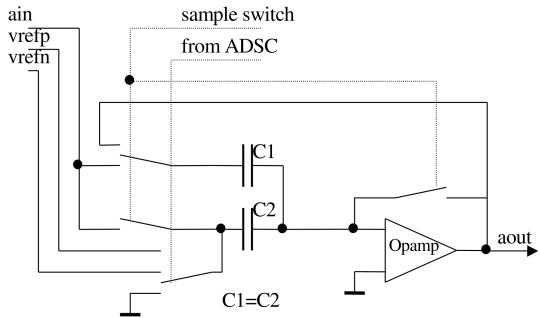

Figure 10: MDAC schematic (single-ended).

MDAC Design cont’

- Fully-differential: invert input to get desired operation

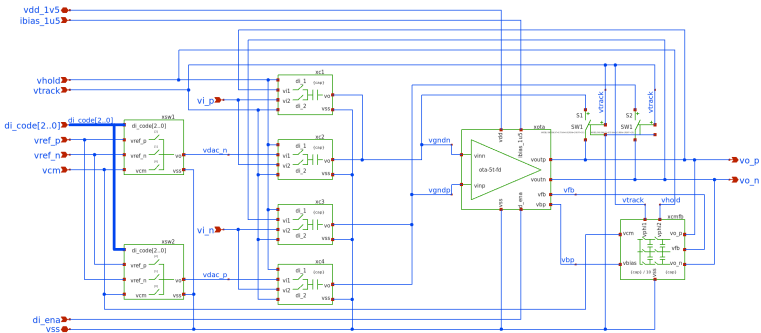

Figure 11: MDAC schematic (fully-differential).

MDAC Design cont'

- $V_o(t) = -V_{step}\frac{1}{\beta}\frac{L_0}{1+L_0}(1 - e^{-t/\tau_o}) \quad \rightarrow \quad \epsilon_d = e^{-t/\tau_o}$
- $\tau_o \approx \omega_{ug}$
- Settling accuracy: $\epsilon_{d,target} = 1/2^{n-n_1} = 2^{-2} = 0.25$
- $T_{settle} = 1/f_s/2$
- $N_{settling} = \frac{T_{settle}}{\tau_o} = -\ln(\epsilon_{d,target}) = 1.4 \quad \rightarrow \quad 2$
- $C_1 = C_2 = 100\,fF$ (sufficient in terms of kT/C noise for 3-bit)
- DC gain from static error
- Slew-rate: $SR = V_{o,step}/T_{settle} = I_d/C_{load}$
- $\rightarrow$ use gm/Id design script for OTA design

MDAC Design cont’

- Keep it simple: basic 5T OTA w/ SC-CMFB (account for additional loading)

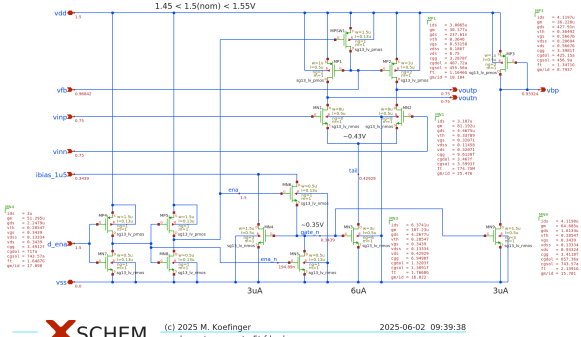

Figure 12: 5T-OTA schematic (ac-params annotated)

Circuit Level Simulation

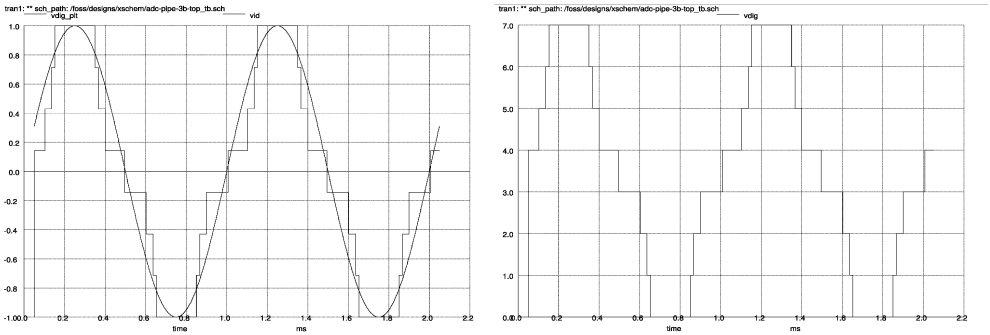

Figure 13: $A_{\mathrm{sig}} = 1\,\mathrm{V_{peak}}$ $\mathrm{f_{sig}} = 1\,\mathrm{kHz}$

Circuit Level Simulation

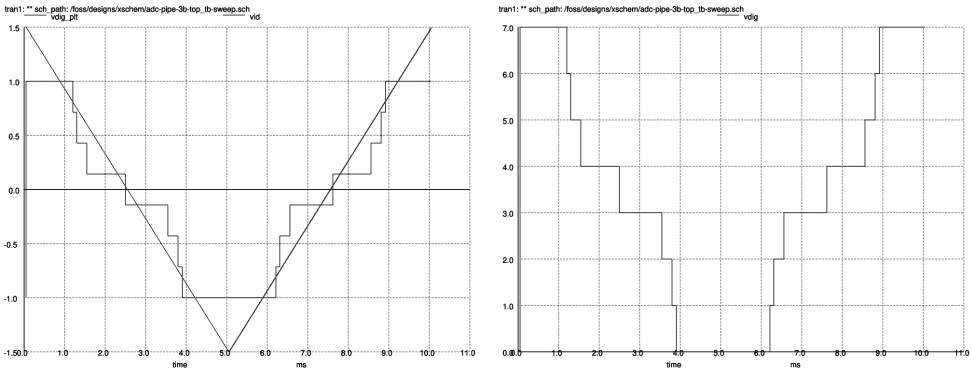

Figure 14: 100 Hz triangle input and scaled code; code only

Summary and Known Issues

- Gain error
- ADC macro only handles $\mathrm{f_{sig}} < \mathrm{f_s}/10$
- Redesign OTA for more gain (different architecture e.g. two-stage, folded-cascode)
- INL, DNL calculation is missing