

SRAM Compiler Design Plan v0.1

RIOS Lab

2023-01

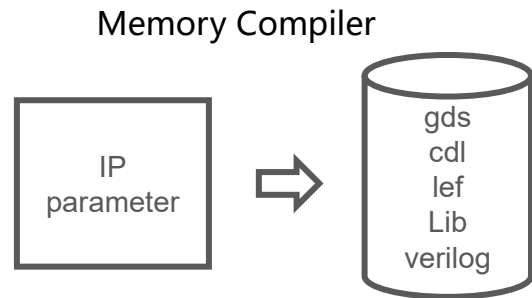
Memory Compiler (AI based)

- **Key Feature**

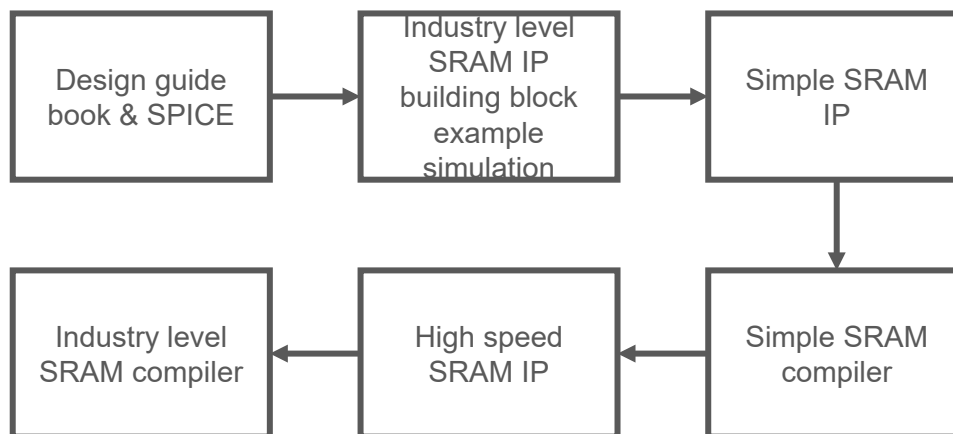
- Advanced read / write assistant for advanced process node(beyond 7nm)
- First open-source high speed SRAM memory compiler

- **Challenge**

- Mixed-signal (digital+analog) high speed design
- May need fast spice simulator



Design Stage



1 Introduction

2 IC design flow with EDA

2.1 Design flow

2.2 SPICE language

3 PDK and standard cell library

4 SRAM architecture

4.1 Simple architecture

4.2 Full typical architecture

5 SRAM timing

6 SRAM building block

6.1 Array

6.2 Decoder

6.2.1 WL decoder&driver

6.2.2 WL control (timing)

6.2.3 Pre-charge

6.2.4 Column mux

6.2.5 Pre-decoder

6.3 Read and write

6.3.1 Sense amp

6.3.2 Write driver

6.3.3 Datapath

6.4 Control

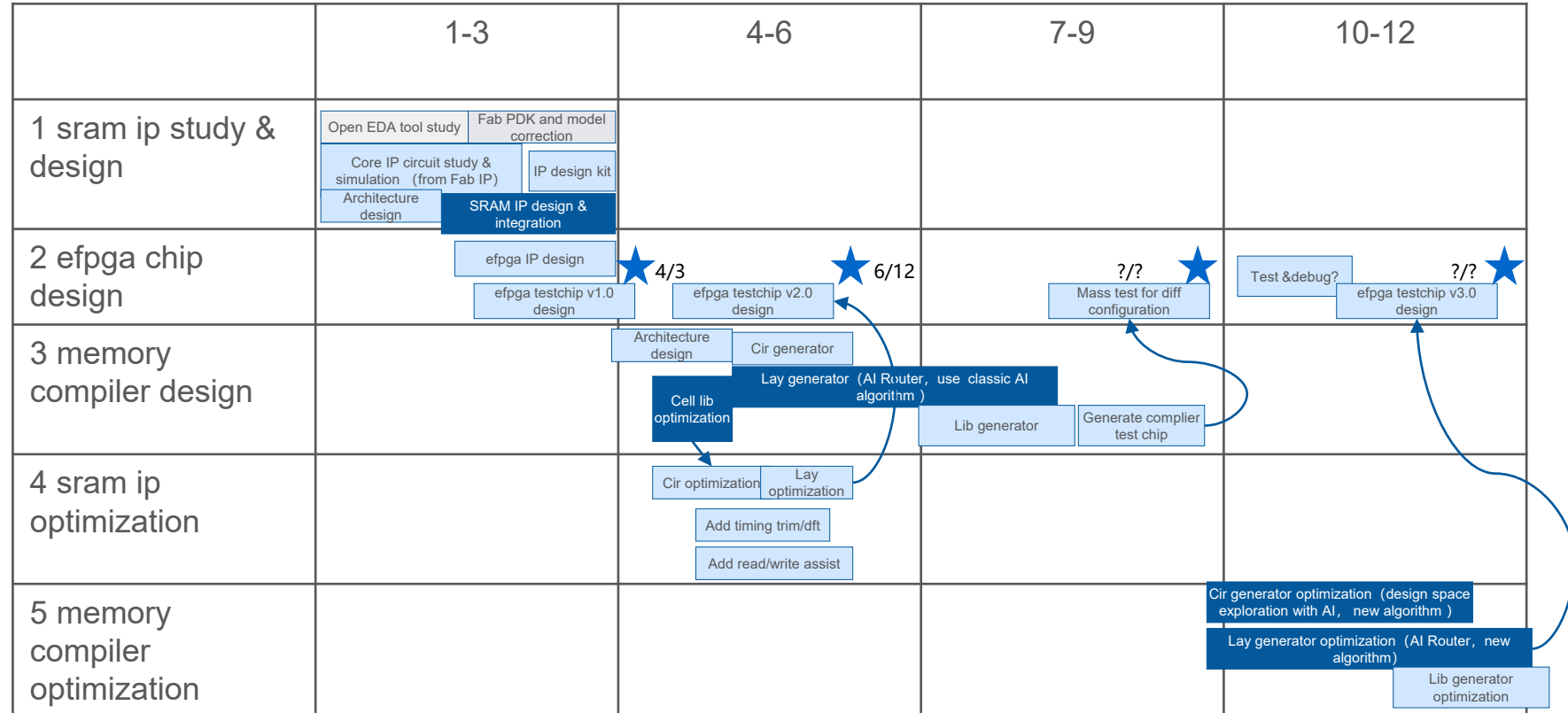
6.4.1 Control Unit

6.4.2 Timer

6.5 DFT (Optional)

7 Verification

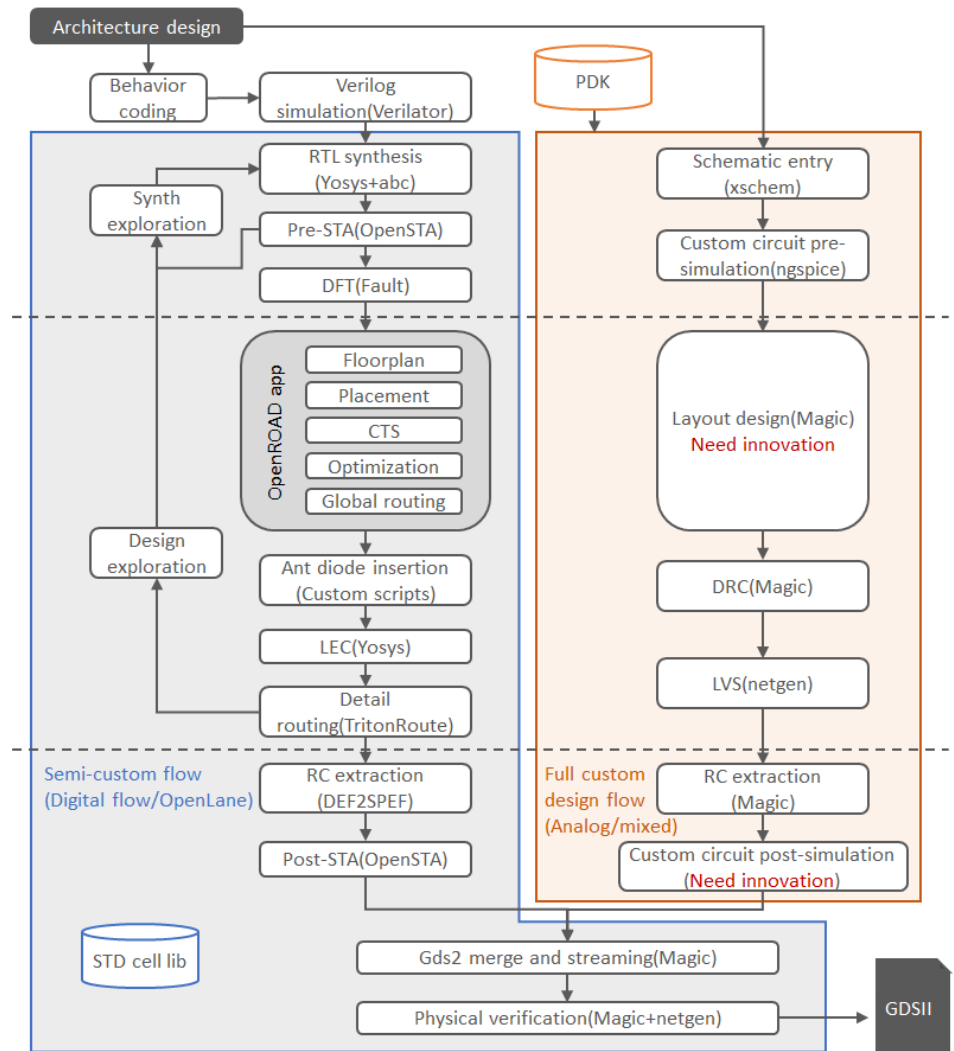
Memory Compiler (AI based) Roadmap



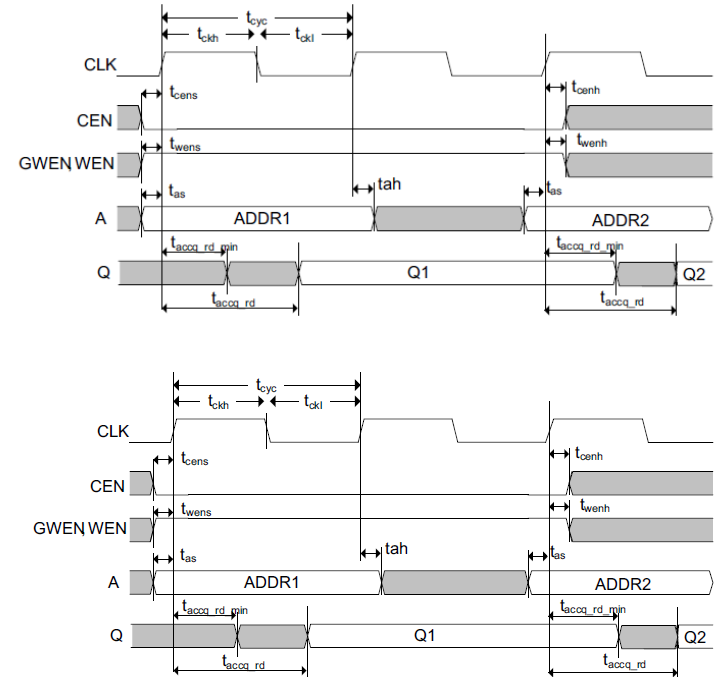
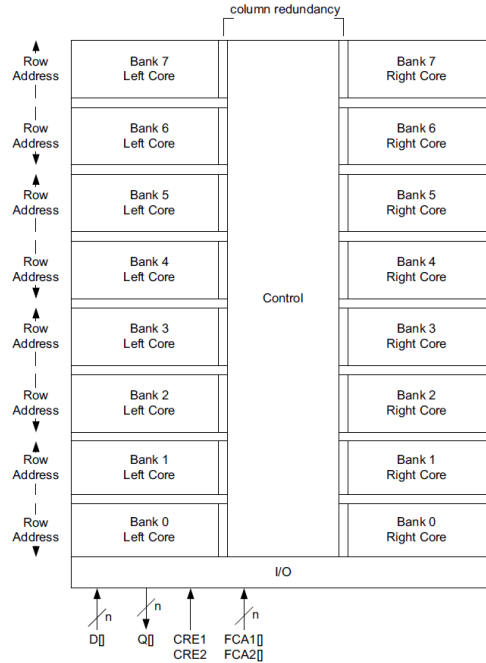
EDA Design Flow

• Design Flow

- Design with SPICE simulator
- Mixed-signal design flow
- OpenLane
- Full custom design → Automation generator



Multi-bank and Timing



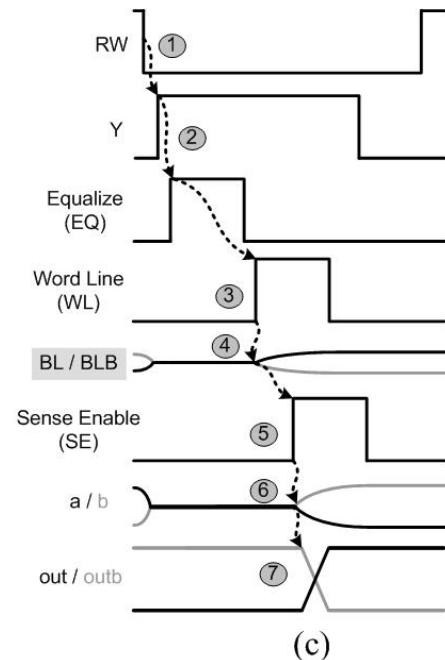
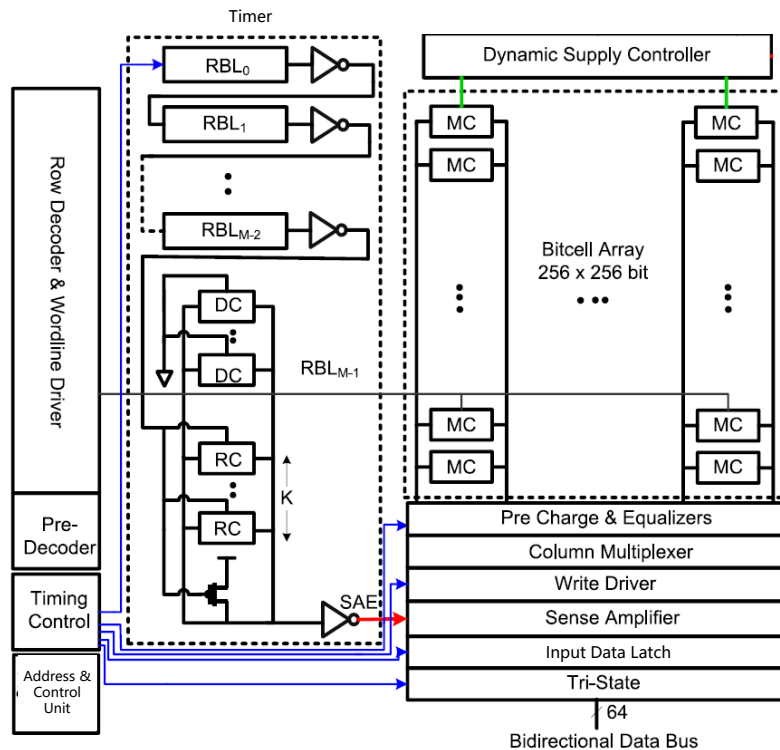
Multi-bank architecture to enhance performance

SRAM Base Architecture

• Build Block

- Array
- Decoder
 - WL decoder & driver
 - WL control (timing)
 - Pre-charge
 - Column mux
 - Pre-decoder
- Read and write
 - Sense amp
 - Write driver
 - Datapath
- Control
 - Control Unit
 - Timer
- DFT (Optional)

• Timing



THANKS