Design and Demonstration of an 8-bit Bit-Serial RSFQ Microprocessor: CORE e4

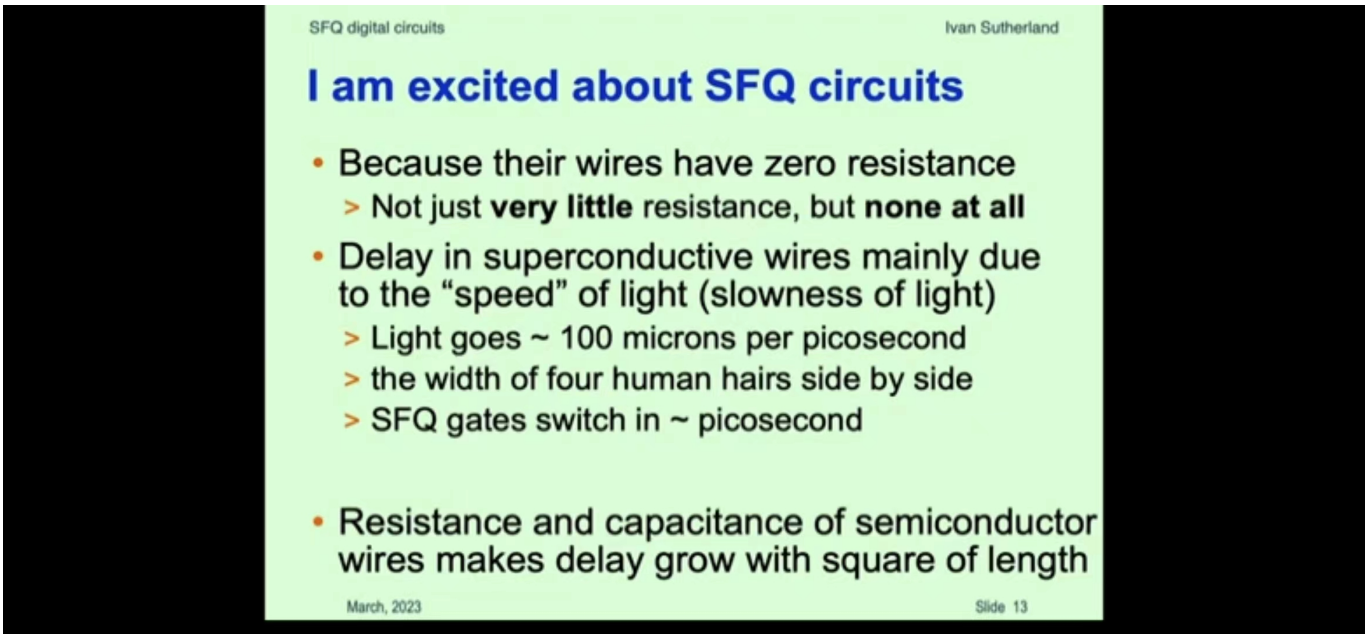
Design of an 8-bit Bit-Parallel RSFQ Microprocessor

> “We have designed, fabricated, and tested the first version of an 8-bit bit-serial RSFQ microprocessor, which is called CORE e4.”

Basic Concepts

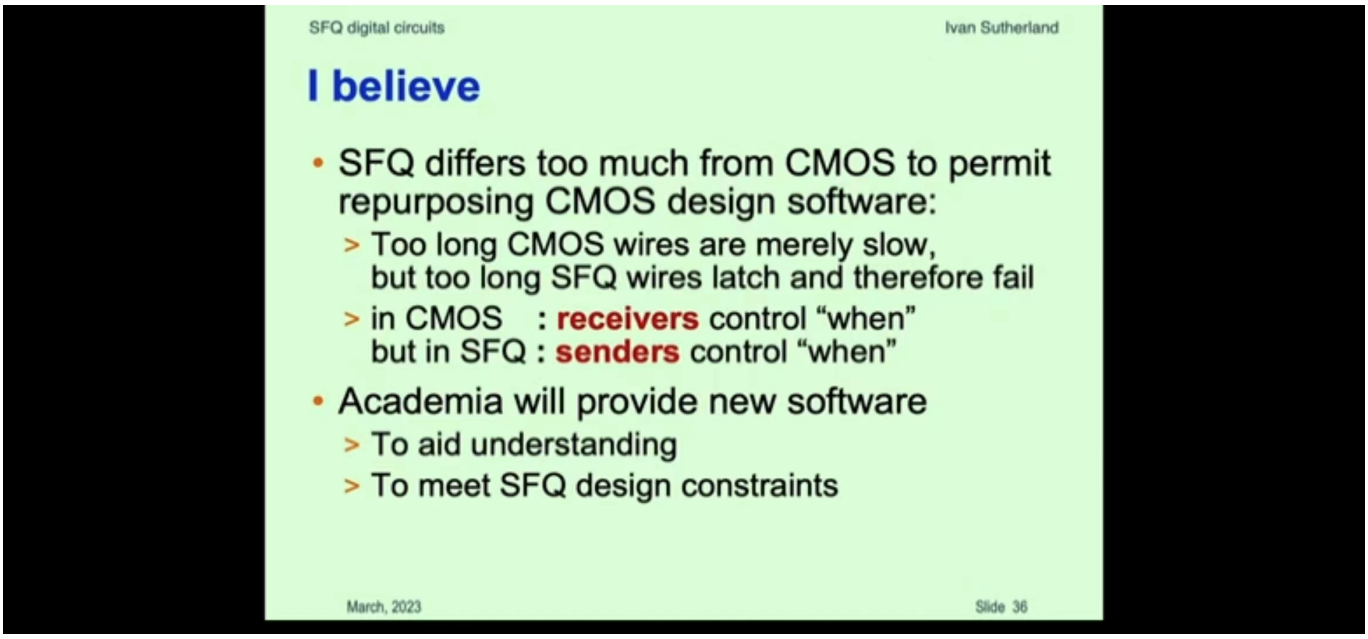
RSFQ

> 在电子学领域中，快速单通量量子（rapid single flux quantum，缩写RSFQ；也称快单磁通量子）是一种使用超导设备的数字产品，也称约瑟夫森结, 用来处理数字信号。在RSFQ逻辑中，信息以磁通量量子的形式存储，并以单通量量子（Single Flux Quantum，缩写SFQ）电压脉冲的形式传递。



- 与一般计算机使用的CMOS晶体管技术有很大差异：

    1. 超导设备需要低温；
    2. 约瑟夫森结产生的皮秒级SFQ电压脉冲被用于编码、处理和传输数字信息，这代替半导体电子器件中晶体管产生的电压电平；
    3. SFQ电压脉冲在超导传输线上的进程非常小，并且如果脉冲的频谱分量不超过超导体能隙的频率，通常可以忽略不计；
    4. 在1ps的SFQ脉冲情况下，电路计时可能达到100 GHz频率（每10皮秒一次脉冲）



Josephson Junctions

> **库珀对**：在低温超导体中，电子并不是单个地进行运动，而是以弱耦合形式形成配对，一般称之为库珀对．形成库珀对的两个电子，一个自旋向上，另一个自旋向下。

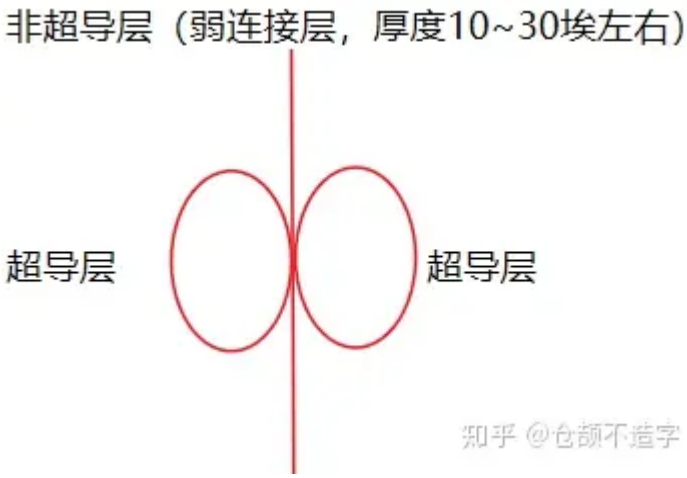
> **隧道效应**: 隧道效应由微观粒子波动性所确定的量子效应。又称势垒贯穿。考虑粒子运动遇到一个高于粒子能量的势垒，按照经典力学，粒子是不可能越过势垒的；按照量子力学可以解出除了在势垒处的反射外，还有透过势垒的波函数，这表明在势垒的另一边，粒子具有一定的概率，粒子贯穿势垒。
> 约瑟夫森效应属于隧穿效应，但有别于一般的隧道效应，它是库伯电子对通过由超导体间通过弱连接形成约瑟夫森结的超流效应。

> **临界电流**：环境中无电场磁场时，结的两端没有电势差，库珀对发生隧穿现象，因此约瑟夫森结中出现了直流电流(超导隧道电流),这个电流存在最大值 IC

- 什么是约瑟夫森结?

  约瑟夫森结实际上是一种电子电路，由两个超导层和一个非超导层紧密连接，非超导层厚度为纳米级别。

  这个非超导层对于电子来说就是'势垒'，温度足够低的时候，超导体快速通过势垒交换成对电子，产生量



  子隧道效应。

- 约瑟夫森效应 电子从绝缘层的一侧超导体通过隧穿运动到另一次超导体, 势能不会降低的反应, 即电子可以通过两块超导体之间绝缘层的量子隧道效应

1. 直流约瑟夫森效应
   结两端电压V=0 时，由于库伯对的隧穿效应，结中会存在超导电流。超导电流小于某一临界电流，那么结始终会保持零电压现象，也就是说结中无电阻。

2. 交流约瑟夫森效应
   约瑟夫森结两端施加直流电压（V≠0）时（产生的电流大于临界电流）

   一方面约瑟夫森结中的超导电子对隧道电流变为高频交变超导电流（频率与施加的直流电压成正比），此时结区向外发射同频率的电磁波（施加电压V是几微伏，频率在微波区域；频率在几毫伏，频率在远红外波段）。因此超导隧道结在直流电压作用下，产生超导交流电流，从而辐射电磁波的现象叫做交流约瑟夫森结效应。

   另一方面，在施加直流电压的同时，如果用频率为 f 的电磁场去辐照约瑟夫森结，电磁场对结的直流电流会产生调制作用， 对外表现为 I-V 曲线上会出现精确的直立，电压台阶，这样的台阶称之为“夏皮诺台阶”（Shapiro step），同时 Ic 的大小也有所压缩。电压台阶所对应的电压 Vn 与电磁波的频率 f 成正比，即 Vn=nhf/2e=nΦ0f，式中 n=0, 1, 2…。

   简单来说, 当势垒两边施加一直流电压V时，伴随以频率为2eV/h的超导交流通过势垒，并且这个频率与连接电路所用的具体材料无关。

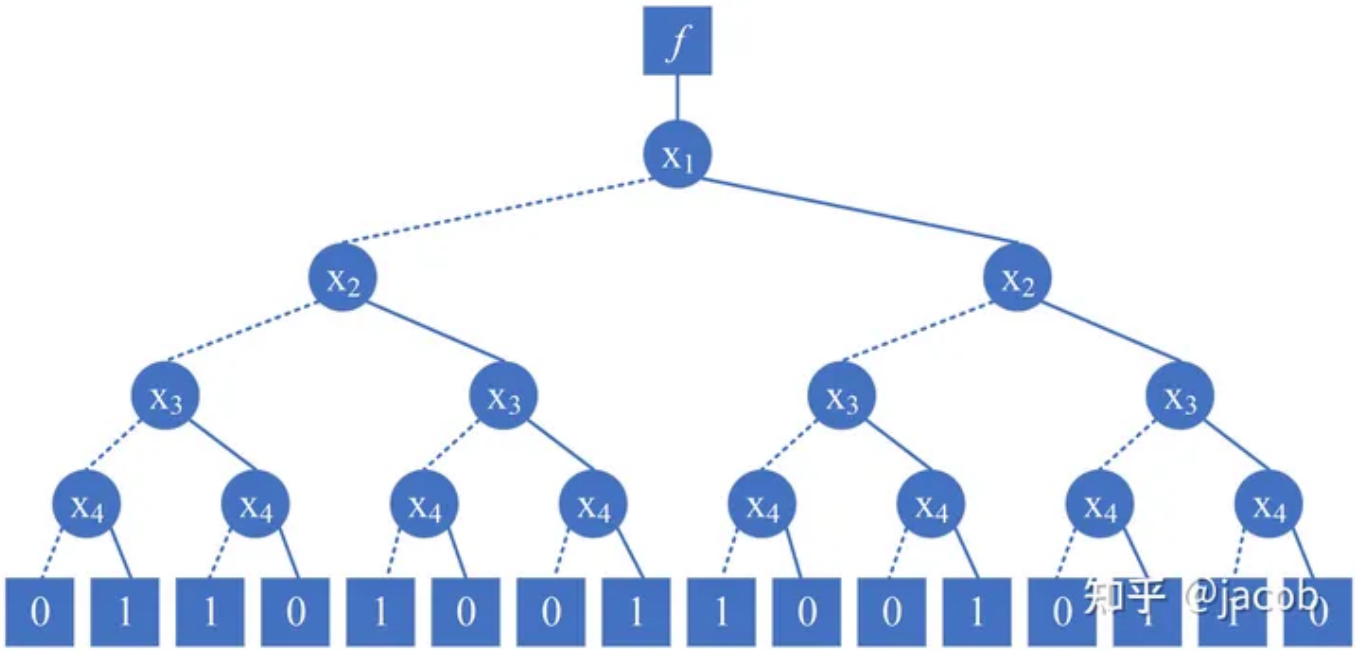
Bias Current 偏置电流

> 晶体管构成的放大器要做到不失真地将信号电压放大，就必须保证晶体管的发射结正偏、集电结反偏。即应该设置它的工作点。所谓工作点就是通过外部电路的设置使晶体管的基极、发射极和集电极处于所要求的电位（可根据计算获得）。这些外部电路就称为偏置电路（可理解为，设置PN结正、反偏的电路），偏置电路向晶体管提供的电流就称为偏置电流。

DC Bias margins(unsolved)

余度: 保证电路的稳定性和安全性

Binary Decision Diagram - BDD



1986年，Bryant发表论文指出归约有序的二元决策图是布尔函数的规范表示。

One-bit Code

常见于机器学习中, 使用one hot编码器对类别进行"二进制化"操作

Content

Introduction

- 现状
    1. SFQ circuit使用超导设备, 可以实现超高速度计算(tens of gigahertz), 能耗低
    2. 目前有不少SFQ logic circuit, 包括reduced-static power RSFQ, zero-static power RSFQ, ac-biased new logic families
    3. FLUX, TIPPY, bit-serial microprocessors based on the complexity-reduced(CORE) architecture: CORE 1 α and CORE 1 β
- Previous RSFQ microprocessors:
    1. No memory to store both instructions and data
    2. Only for primitive instructions
    3. Demonstrated operation frequency was up to 20 GHz

> Our goal is to show that developing a stored-program computer using RSFQ circuits is possible.

Design of CORE e4

Overview

> The microarchitecture of CORE e4 is similar to that od CORE 1 α

- Based on Havard architecture
- The target frequency for the sysclk is 2GHz(吉兆,1Ghz=1000Mhz), target frequency for the bit-serial processing is 50 GHz

- 4 registers are provided, and register indirect access (寄存器间接存取) is adopted to handle sequence data in the memory
- Instruction and data memory have 256 bits(8bits*32words) each and are addressable(指令寻址&数据寻址) by a 5-bit address.

Instructions

TABLE I
INSTRUCTION SET OF THE CORE e4

Instruction	Instruction Word	Definition
ADD	1100bbaa	RegA ← (RegA)+(RegB)
MV	1101bbaa	RegA ← (RegB)
SUB	1110bbaa	RegA ← (RegA)−(RegB)
CMP	1111bbaa	(RegA)−(RegB)
AND	1010bbaa	RegA ← (RegA) AND (RegB)
XOR	1001bbaa	RegA ← (RegA) XOR (RegB)
OR	1011bbaa	RegA ← (RegA) OR (RegB)
NOR	1000bbaa	RegA ← (RegA) NOR (RegB)
INC	000010aa	RegA ← (RegA)+1
DEC	001010aa	RegA ← (RegA)−1
LD	000100aa	RegA ← (Mem[(Reg0)])
ST	000101aa	(Mem[(Reg0)]) ← (RegA)
SKGE	00000100	PC ← PC+1 if greater than or equal to
SKLT	00000101	PC ← PC+1 if less than
SKZ	00000110	PC ← PC+1 if zero
SKNZ	00000111	PC ← PC+1 if not zero
SET	010ddddd	Reg0 ← D
JMP	011ccccc	PC ← C
NOP	00000000	No operation
HLT	00000001	Stop

A=[aa], B=[bb], C=[ccccc], D=[ddddd]

- program status word(PSW) is set after an ALU instruction is executed.

Component Circuit

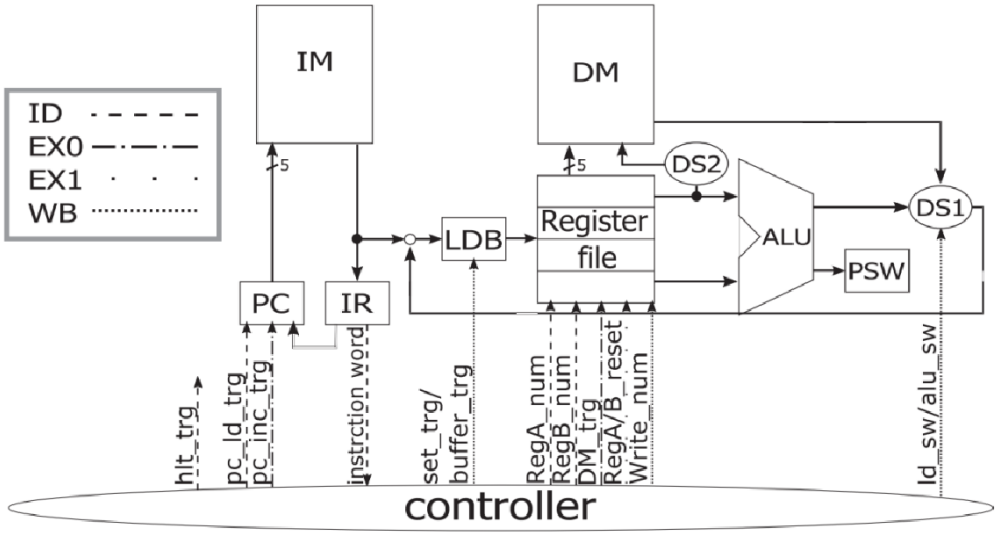


Fig. 1. Microarchitecture of the CORE e4.

Instruction Execution

1. Instruction Fetch 0 (IF0)

   An instruction is read out from the IM according to the value of the PC. The PC is incremented.

2. Instruction Fetch 1 (IF1)

   This phase is reserved as the latency period for instruction propagation delay.

3. Instruction Decode (ID)

   The instruction is decoded at the controller, and the behavior of each component is determined. At ALU instructions, the registers to be read are selected by RegA_num and RegB_num.
   Instruction HLT is executed in this phase (hlt_trg). At instruction JMP, the lower five bits in the IR are loaded to the PC by pc_ld_trg.

4. Execution 0 (EX0)

   At ALU instructions, the content of the registers is read out. At instruction LD, the DM is accessed to obtain the value by DM_trg. At instruction ST, the existing data in the word to be written in the DM are cleared. If the branch condition is satisfied, the PC is incremented to skip the next instruction by pc_inc_trg.

5. Execution 1 (EX1)

   At ALU instructions, calculation in the ALU is completed and the registers to be read are deselected by

   RegA/B_reset. At instruction LD, this phase is the latency period for data transfer. At instruction ST, data are transferred from the register to the DM through DS2.

6. Write Back (WB)

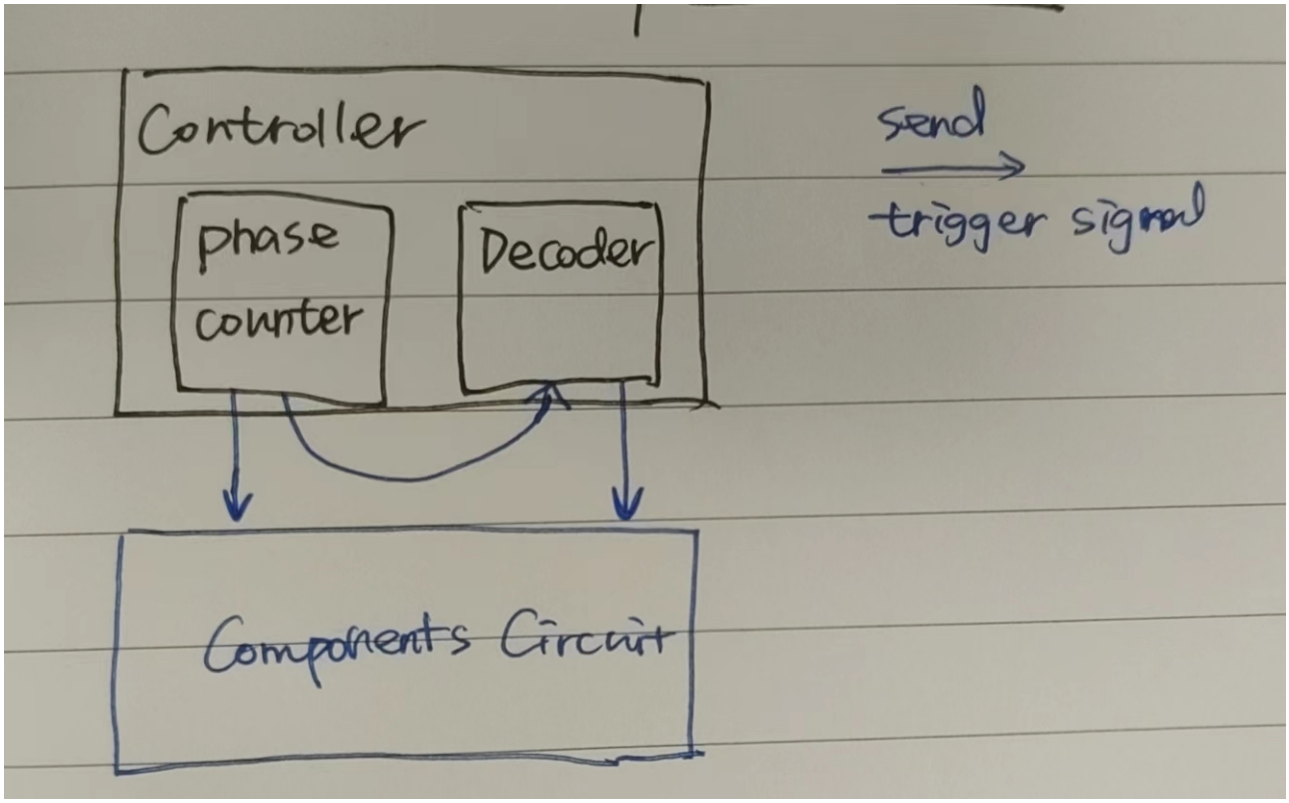
   At ALU instructions, as well as instructions SET and LD, the register to be written is selected by Write_num. At ALU instructions, alu_sw is sent to DS1, and the calculation result of the ALU is sent to

the LDB. The value of the LDB is then written to the register by buffer_trg. At instruction LD, ld_sw is sent to DS1, and the value of the DM is sent to the LDB. The value of the LDB is then written to the register by buffer_trg. At instruction SET, the lower five bits of the LDB are written to the register by set_trg.

Design

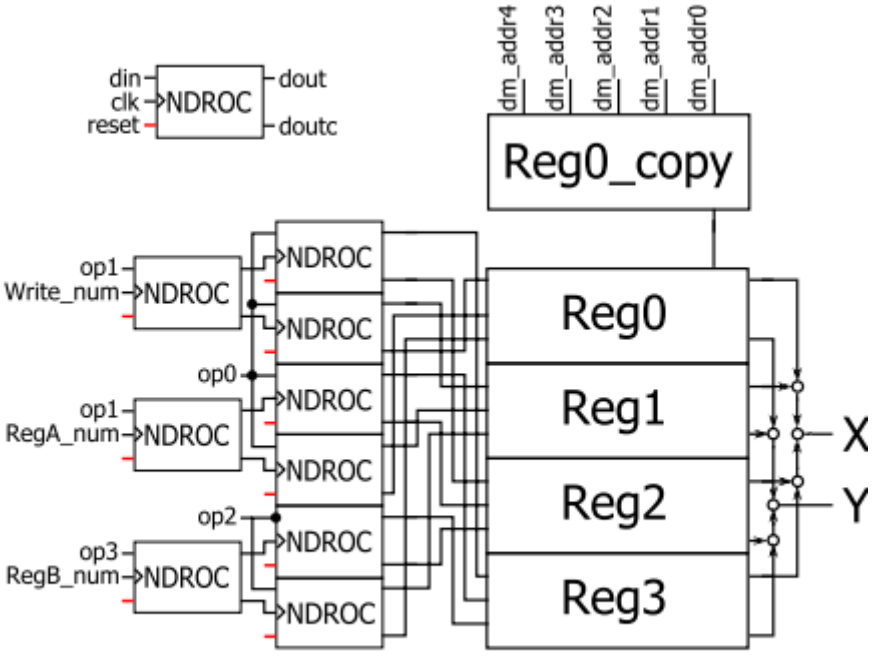
Controller

- consists of a phase counter and a decoder



Register File

- 4 shift registers(Reg0, Reg1, Reg3, Reg3), each of which holds an 8-bit serial data word
- buffer: hold a copy of Reg0, can be read out parallel(shorten the access time to the DM)



- **What is NDROC?** Fig. 2. Register File of the CORE e4.

ALU

- The operation to be executed is determined by 4-bit control signals.
- The ALU performs the decoding of 4-bit control signals and ALU operations.

- $alu1 = op7;$
- $alu2 = op6 \vee INC \vee DEC;$
- $alu3 = op5;$
- $alu4 = op4.$

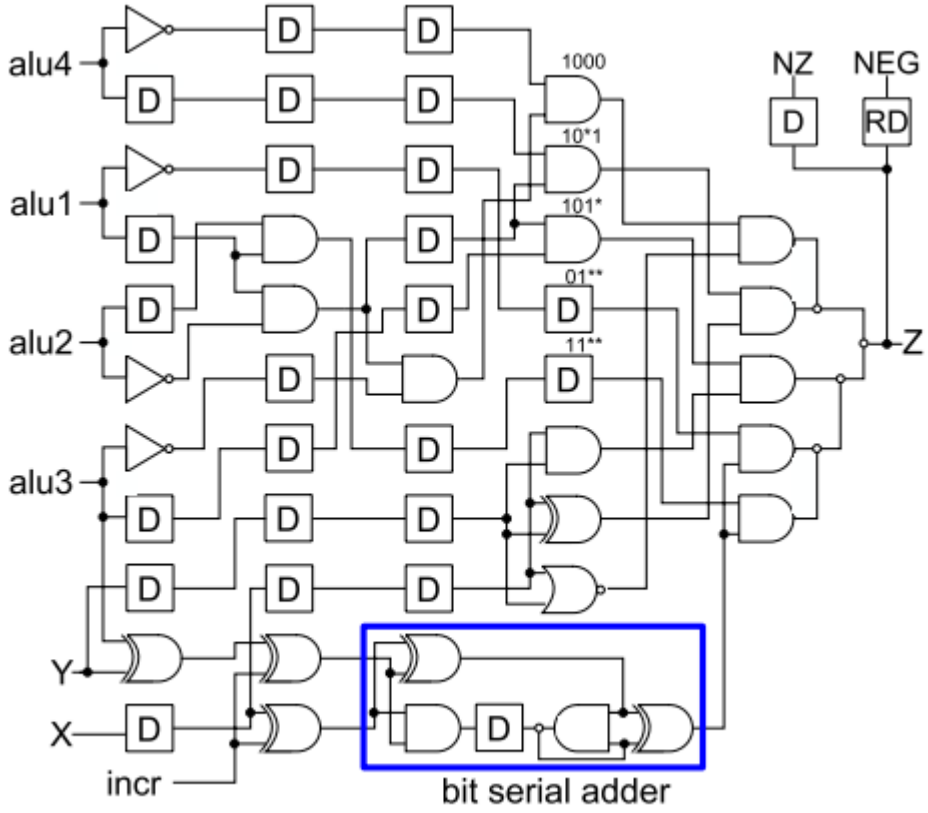


Fig. 3. ALUv1 of the CORE e4.

Simulation

> "As a result of the measurement, we successfully obtained the correct operation of ten ALU instructions. The estimated maximum operating frequency based on measurement results is 80 GHz."

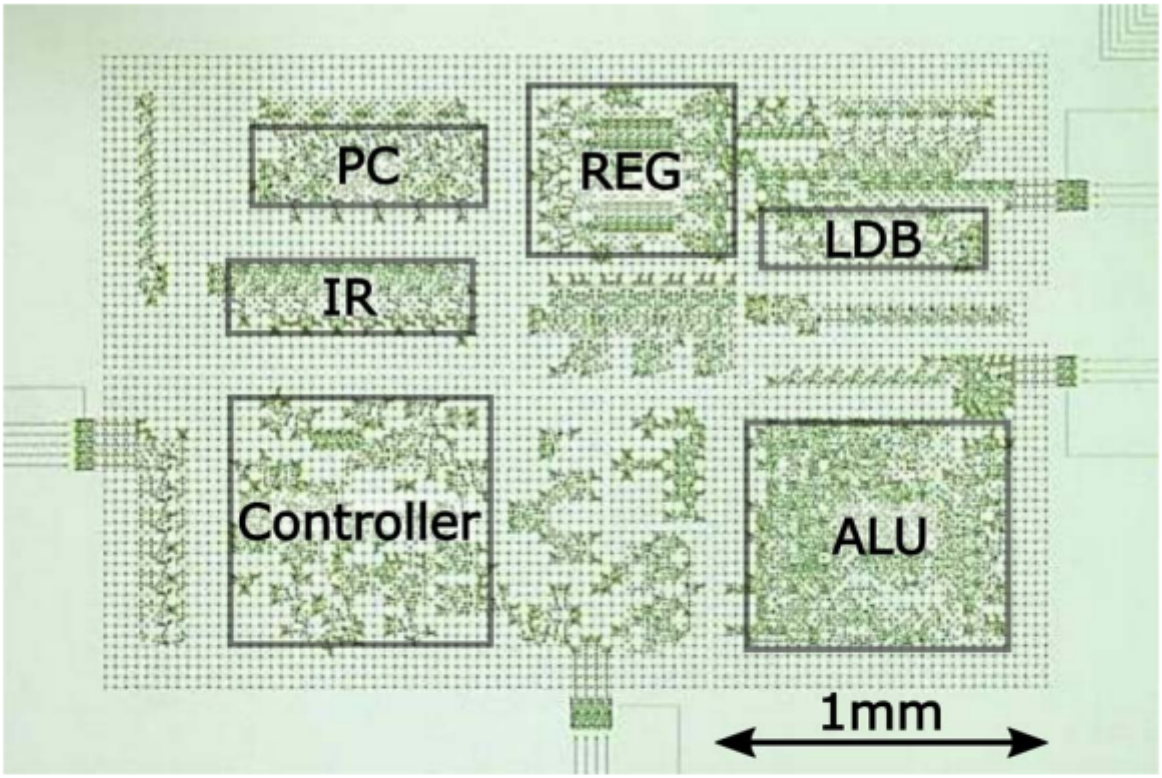


Fig. 4. Micrograph of the CORE e4v1.

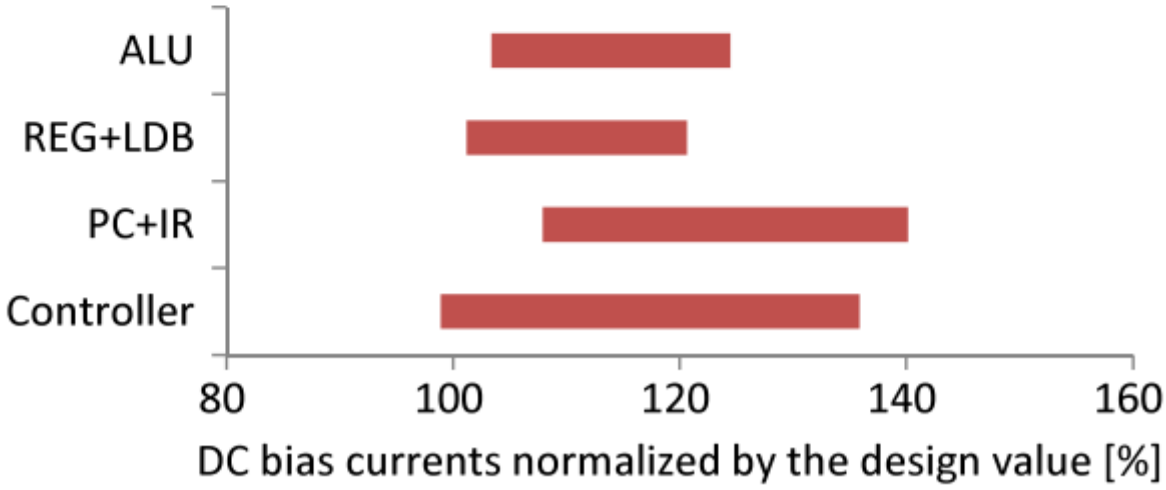


Fig. 5. Bias margins at instruction MV.

- overlap more than 10% -> ensure the scurity and stability

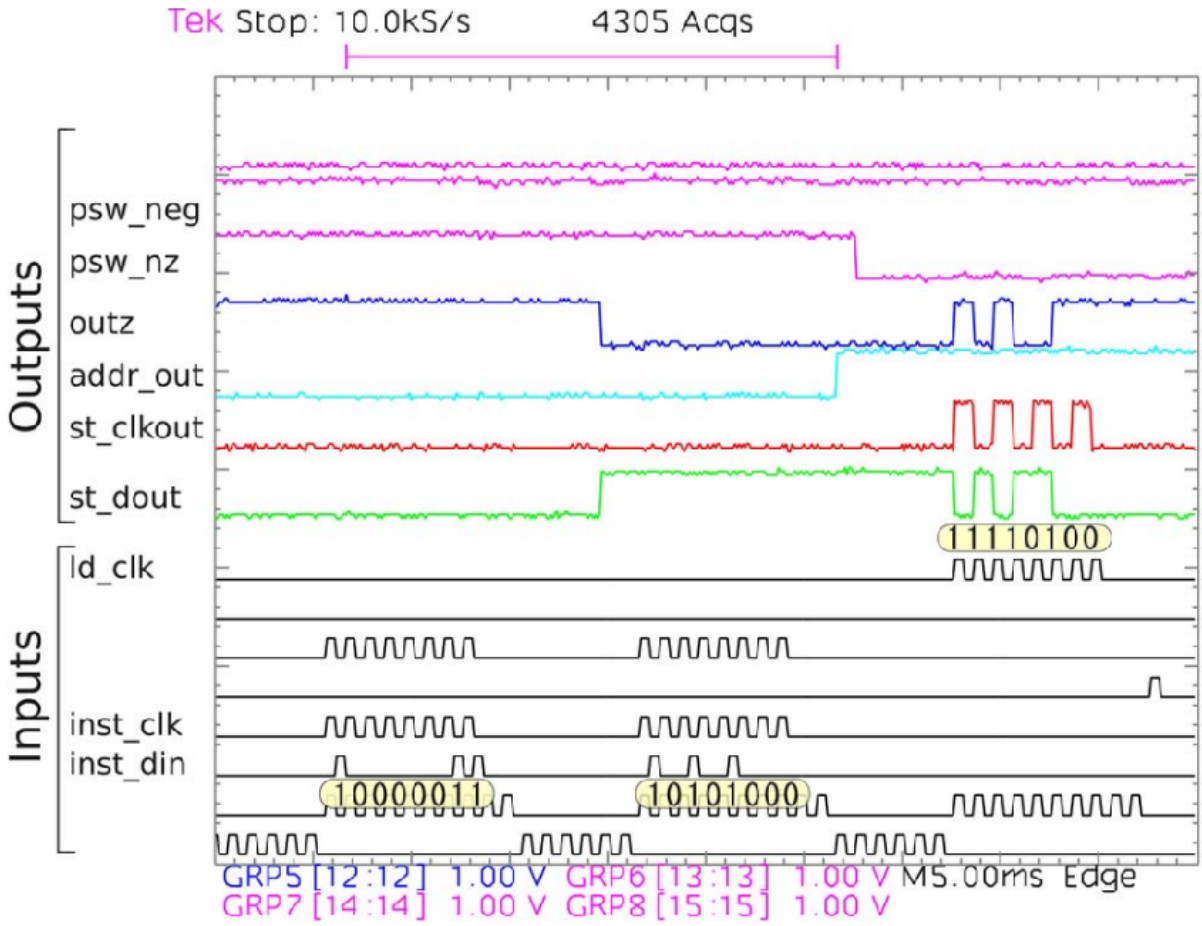


Fig. 6. Input and output waveforms of the LD-LD-ADD-ST instructions.

> Thirteen of the 20 instructions(ADD, MV, SUB, CMP, AND, XOR, OR, NOR, SET, LD, ST, NOP, and HLT) were successfully demonstrated in our measurement. However, the output lines labeled addr_out and the operation of INC/DEC instructions were not successfully demonstrated because of errors in the clock wiring.

Reference

- Design and demonstration of an 8-bit bit-serial RSFQ microprocessor: CORE e4
- 二元决策图(Binary Decision Diagrams - BDD)（一） - jacob的文章 - 知乎 https://zhuanlan.zhihu.com/p/397164596

- https://www.youtube.com/watch?v=KMVV3ErGSVY