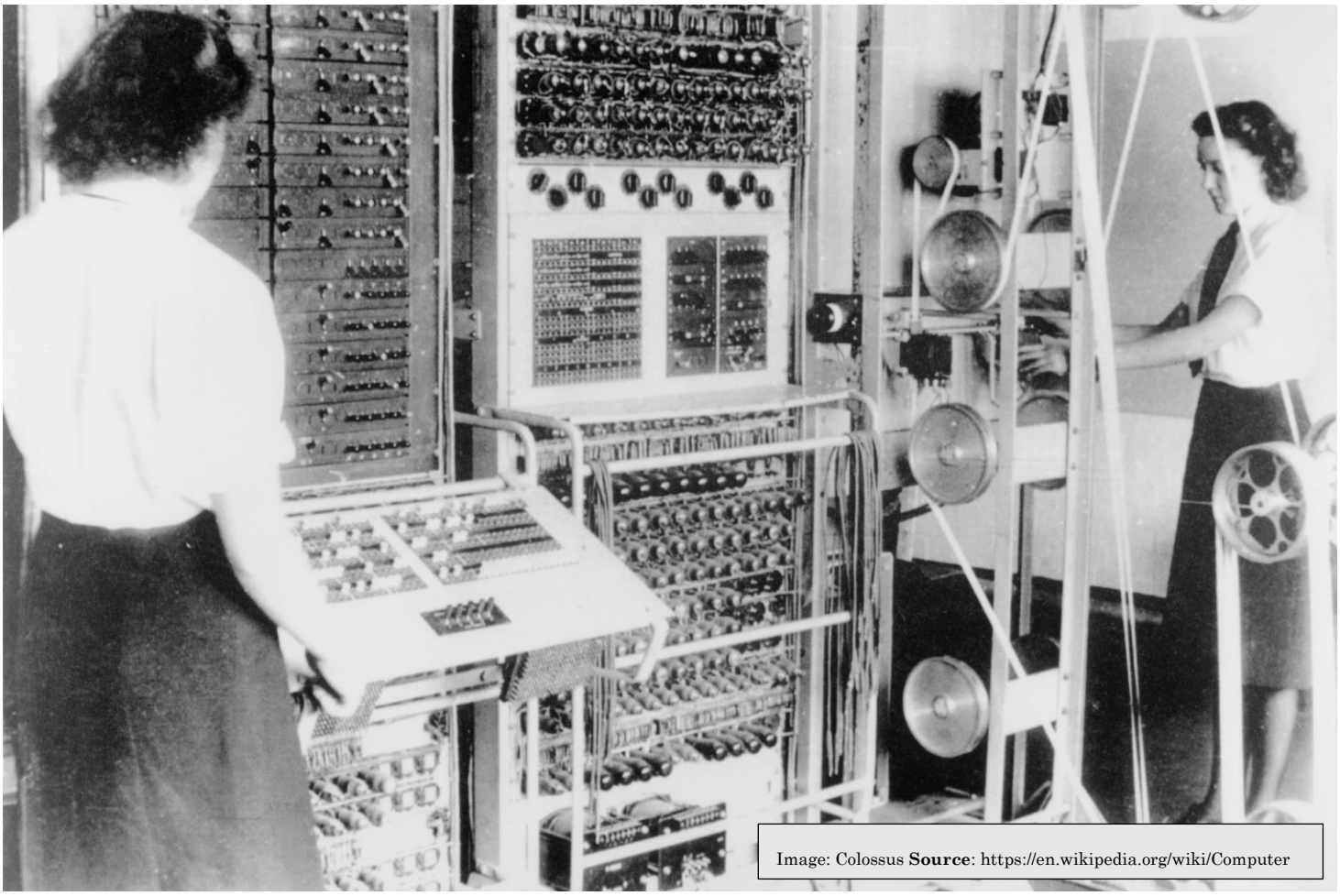

Image: Colossus **Source**: https://en.wikipedia.org/wiki/Computer

DEU Electronic Universal Automatic Reduced Computer (DEUARC) Design

CME 2206 Computer Architecture 2022-2023 Term Project

CME 2206 – LAB PROJECT

DESCRIPTION

You will design a basic computer that called DEUARC (**D**EU **E**lectronic **Uni**versal **A**utomatic **R**educed **C**omputer). DEUARC has 9 registers, 3 memories, arithmetic and logic unit, control unit and bus system.

Quartus II software will be used to design and verify DEUARC. The project is given as a term project and will be implemented in weekly lab sessions. It is advised that you read problem definitions of all of them before actually starting to implement your design, i.e., Common Bus and Registers.

Please submit zipped All Files (don't forget to submit waveform of the results) of the simulations for each lab session.

GENERAL STRUCTURE OF DEUARC

REGISTERS

DEUARC has 9 registers which are *Address Register, Program Counter, Stack Pointer, Input Register, Output Register, Instruction Register* and *3* general purpose registers.

MEMORIES

In DEUARC, there are two memories, which are ***instruction (32x11), data (16x4) and stack (16x5)*** Each has "read enable" signals and "data inputs". *Data and stack memory* also has "write enable input".

COMMON BUS SYSTEM

Common bus system will be responsible for data flow and provide data transfer between register and/or memories.

ARITHMETIC AND LOGIC UNIT

In ALU, arithmetic and logical operations will be held.

CONTROL UNIT

Control unit processes instructions to direct the micro-operations for computer's memories, registers and arithmetic/logic unit. Control unit consists of decoders and a number of control logic gates. It should produce operation signals and time periods for fetching, decoding and executing the instructions.

ASSIGNMENT 3 – CONTROL UNIT

DEUARC has three instruction code formats as shown in the **Hata! Başvuru kaynağı bulunamadı.**. The type of the instruction recognized by the computer control unit using four-bits opcodes. You should generate a list for the control function and microoperations of DEUARC (as Table 5.6 of Mano's Basic Computer) before designing control unit. Control unit includes logical designs to control registers, memories, common bus and ALU. Figure 1 and Figure 2 show general view of control unit design and DEUARC respectively. **But they haven't to be completed or correct, so you may add or change signals, components etc. in your designs.**

Table 1 - DEUARC Instruction Set

Symbol		Description
Operation	***Opcode***	
HLT	0111	Halt the computer
Arithmetic and Logic Operations		
Q(1 bit) | Opcode (4 bits) | Rd (2 bits) | S1 (2 bits) | S2 (2 bits)		
DBL	0000	Double content of S1 and store result in Rd
DBT	0001	Divide content of S1 by 2 and store result to Rd
ADD	0010	Add content of S1 and S2 and store result in Rd
INC	0011	Increase content of S1 and store result in Rd
AND	0100	AND contents of S1 and S2 and store result in Rd
NOT	0101	Complement S1 content and load the result into D
XOR	0110	XOR contents of S1 and S2 and store result in Rd
Data Transfer		Q(1 bit) | Opcode (4 bits) | Rd (2 bits) | S1 (2 bits) | S2 (2 bits)
ST	1000	writing the content of Rd into the memory of address S1S2 if Q=0 writing the data *S1S2* into the memory of address *the content* of *register Rd* if Q=1
LD	1001	reading the memory content of address *S1S2* and load it into Rd, if Q=1 reading the data *S1S2* and load it into Rd, if Q=0
IO	1010	Transfers data from S1 into OUTR, if Q=0 Registers: 00→ R0, 01→ R1, 10→ R2 Transfers data from INPR into Rd, if Q=1 Registers: 00→ R0, 01→ R1, 10→ R2
TSF	1011	Transfers data from S1 into Rd. Registers: 00→ R0, 01→ R1, 10→ R2
	Registers	00→ R0, 01→ R1, 10→ R2
Program Control		- (1 bit) | Opcode (4 bits) | -(1 bit) | Address (5 bits)
JMP	1100	if Q=0 then jumps to address (5-bits) if Q=1 and if V=1 then jumps to address (5-bits)
CAL	1101	go to the address of the instruction memory (PUSH operation of stack memory)
RET	1110	load the previous PC content from the stack into PC (POP operation of stack memory)
JMR	1111	- (1 bit) | Opcode (4 bits) | - (2 bits) | Address (4 bits - signed) Use Address as offset and jump to address relatively

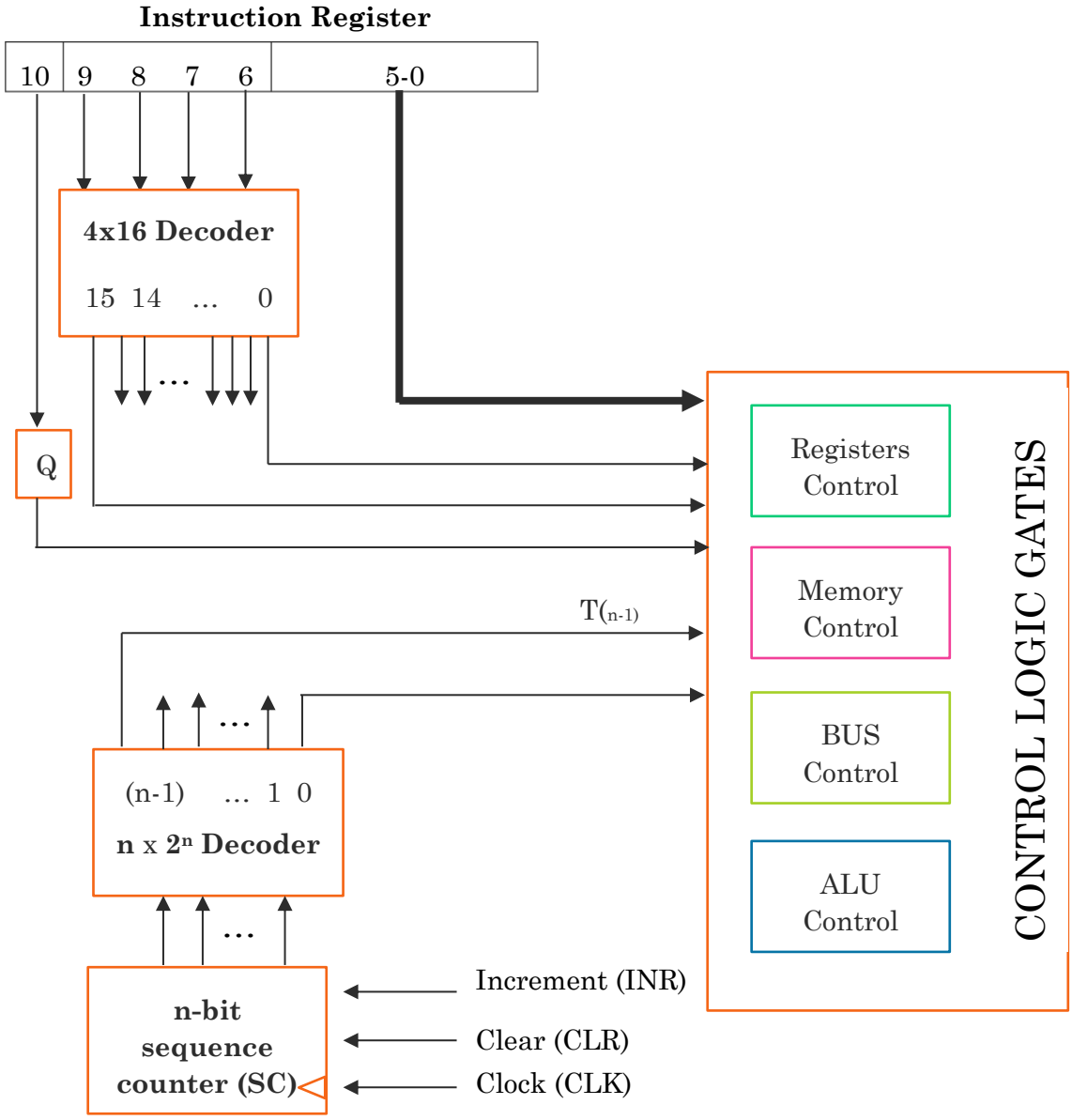

Figure 1 - General structure of Control Unit in DEUARC

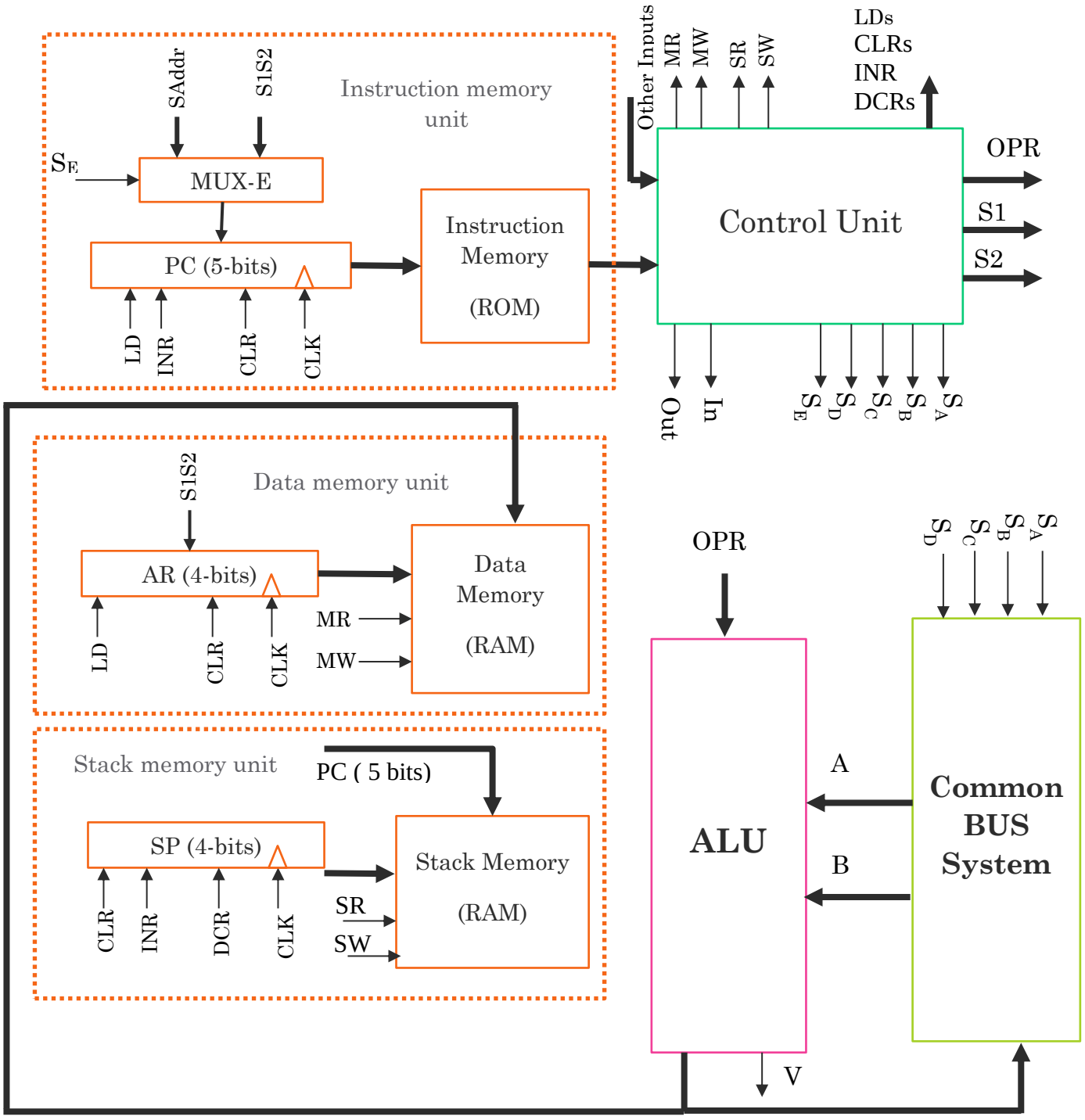

Figure 2 - General view of DEUARC