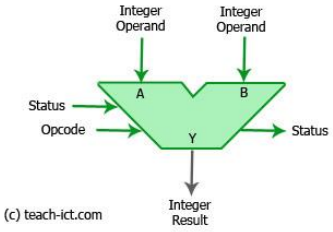
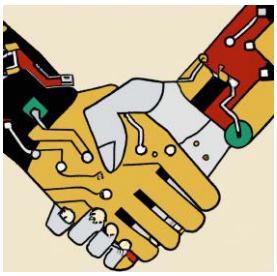
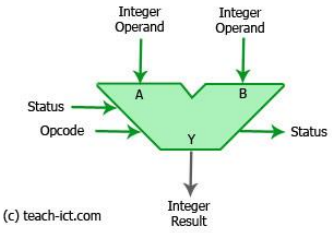
Side channel-protected ALU







Background: Side channel attacks exploit side effects of an algorithm execution to reveal secrets. For example, an implementation of a cryptographic encryption algorithm in hardware consumes power, which may be correlated with the encryption key. The adversary can collect power traces of multiple encryption executions and reveal the key using statistical methods. To counter such attacks, masked hardware implementations have been introduced. In such implementations, a random mask is applied to the processed data to decorrelate the power consumption from the secret.
However, when the algorithm is implemented in software, it uses generic CPU resources, which leak information via the side channel. To solve this problem, we would like to have a 'protected CPU' that can perform operations on masked data. In this project, we will implement a 'protected ALU'.

Project Description: In this project we will implement a protected ALU based on the paper by H. Gross "Sharing is Caring — On the Protection of Arithmetic
Logic Units against Passive Physical Attacks". The implementation will be done in System Verilog and will be synthesized using Synopsys tools. The resulting module will be verified in simulation and the overhead in area and performance will be estimated. Finally, using a simple power consumption emulation technique, the circuit's protection against attacks will be assessed.
The project phases will include:

- Learning the theoretical background of the DPA attacks, the shares-based protection and the specific implementation from the paper.
- Coding the ALU in System Verilog
- Functional verification
- Synthesis and evaluation of the results
- Assessing information leakage using power consumption simulation