Exercise 1 (Allocation of Memory Addresses - Scalar Variables) Consider the following access sequence:

\[ S = \{ b, a, c, a, c, e, c, d, e, d, c, b, a, c, e, c, a \} \]

- Determine the optimal memory address allocation for the five variables \( a, b, c, d, e \), given one address register \( AR0 \) for autoincrement (++) and autodecrement (--) operations.
- List the addressing operations for this optimal memory address allocation and the access sequence \( S \). What is the total cost for the addressing operations?
- Additionally, we have one modification register \( MR0 \) available and can update \( AR0 \) by \(+/-\) the content of \( MR0 \). What is the new total cost for the addressing operations?

Exercise 2 (Allocation of Memory Addresses - Arrays) Consider the following loop with references to the array \( a[] \):

```plaintext
for i=1 to 1024 do
  (1) ref a[i+3]
  (2) ref a[i]
  (3) ref a[i-1]
  (4) ref a[i+1]
  (5) ref a[2*i]
  (6) ref a[2*i-1]
  (7) ref a[i]
endfor
```

- Determine the update values for the references. List the addressing operations, given two address registers. What is the total cost for the addressing operations if autoincrement (++) and autodecrement (--) can be done without additional cost?
- Additionally, we have one modification register \( MR0 \) available. What is the new total cost for the addressing operations?

Exercise 3 (RTG criterion) A hardware designer considers to implement the processor architecture shown in Fig. 1. The architecture contains two functional units, a multiplier and an ALU (for addition and subtraction), as well as four registers \( t, p, R0 \) and \( a \). To enable the development of a good compiler that generates optimal code for tree-like DAGs, the RTG criterion should be checked.

- List all instructions that can be implemented with the given architecture. Group the instructions into MULT-, ADD/SUB-, and transfer -instructions. For each instruction, note the storage locations (registers, memory) for the operands and the result. Operands with constant values can be omitted.
- Construct the register transfer graph and check whether the RTG criterion holds. In case the RTG criterion is not satisfied: Can you satisfy it by removing some instructions? How does the architecture change?
Figure 1: Processor architecture