Aufgabe 1: Integer Linear Programming

Given the sequence graph in Fig. 1.

Figure 1: Sequence graph.

For the execution times of the operations assume: $D_* = 2$, $D_+ = 1$. Two units of the resource type $r_1$ (multiplier) and two units of the resource type $r_2$ (ALU) are allocated.

a) Formulate the problem of latency minimization with restricted resources as an integer linear program (ILP). Hint: Compute the earliest and the latest start times of the operations using the ASAP and the ALAP algorithm, respectively.
b) Give reasons for the latency bound that you used with the ALAP algorithm. How can it be ensured in general that the latency bound is chosen such that the stated optimization model finds a valid solution to the problem of latency minimization with restricted resources?

c) In an analogous manner try to formulate an ILP that solves the problem of cost minimization with latency limitation. Hint: We assume that the cost of a realization is the sum of the costs $c$ of the multipliers with $c(r_1) = 2$ per allocated unit, and of the ALUs with $c(r_2) = 1$ per allocated unit. For the latency bound, we choose $\bar{L} = 6$.

**Aufgabe 2: Iterative Algorithms**

Please answer the following questions considering the given video codec application specified as a marked graph:

![Figure 2: Video codec marked graph representation](image)

![Table 1: Execution time of each function](image)

a) Formulate all existing dependencies in Figure 2 from $\nu_i$ to $\nu_j$ in the form of

$$\tau(\nu_j) - \tau(\nu_i) \geq w(\nu_i) - d_{ij} \times P,$$

where $P$ is the minimum iteration interval.

b) Assuming unlimited resource, determine the minimum iteration interval $P$ and the latency $L$ without pipelining. To justify your answer, draw the scheduling on the timeline given in Figure 3 with the dependency from $\nu_5$ to $\nu_1$ highlighted.
c) The motion estimation function ($v_1$) uses the result of the previous frame (See the dependency between $v_1$ and $v_5$). Let us now suppose that any arbitrary number of tokens can be inserted to reduce $P$ using functional pipelining. Then, how many tokens should be added on the dependency $v_5 \rightarrow v_1$ to achieve $P = 10$? To justify your answer, draw the pipelined scheduling on the timeline given in Figure 4 with the dependency from $v_5$ to $v_1$ highlighted and calculate the latency $L$ of the schedule.