

Scheduling

The scheduler must allocate Instruction & Data Registers and Computational Resources while meeting real-time constraints and minimizing power.

Exposed Architecture

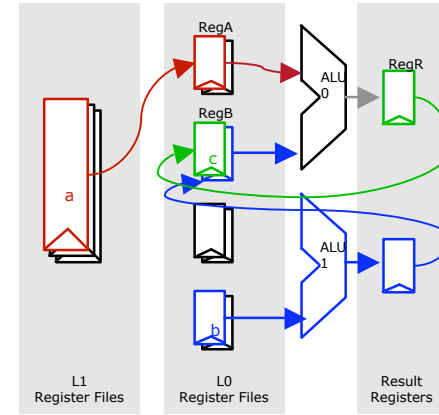
The exposed pipeline gives the scheduler complete freedom in choosing how to use the computation and storage resources, but this flexibility comes at the price of much more complicated scheduling.

Unified Path-based Scheduling

The EEC scheduler searches through time (the schedule) and space (operation & data placement) to find the best path for each operand. The paths are evaluated based on resource utilization and energy cost. This approach schedules the L0 Operand Registers together with the operations in a single phase.

Example 1: $c = a \times b$

The scheduler finds source operand locations and schedules data movement to get operands to correct L0 registers for the chosen ALU. The b operand requires a copy operation to get it from ALU1's L0 register file to ALU0.



Data movement for $c = a \times b$.

Cycle	ALU0				ALU1			L1
	ALU	RegA	RegB	Reg R	ALU	RegB	Reg R	
		0	0	1		0		0
0					MOV	R	W	
1		W		W			R	R
2	MUL	R		R	W			
3			W		R			

Exposed pipeline schedule for $c = a \times b$. R=Read, W=Write.

Example 2: unrolled FIR

The manually unrolled FIR takes maximum advantage of reuse through the L0 Operand Registers. The optimal schedule sustains 8 MACs every 4 cycles, for the entire loop via zero-overhead loops.

The scheduler must detect the reuse of the intermediate values and schedule them in the L0 registers to minimize power and maximize performance.

```

forever {
  sA = __readComm("NORTH");
  sB = sA;

  f0 += a*sA;          f1 += b*sA;
  f2 += c*sB;          f3 += d*sB;

  a = c;
  c = __readComm("NORTH");
  sA = __readComm("NORTH");
  sB = sA;

  f0 += b*sA;          f1 += a*sA;
  f2 += d*sB;          f3 += c*sB;

  b = d;
  d = __readComm("NORTH")
}
    
```

cycle	ALU0						ALU1						LCOM0	SEQ
	ACC0	ACC1	ALU	RegA0	RegA1	RegB0	ACC0	ACC1	ALU	RegA0	RegA1	RegB0		
0						W : sA						W : sB	READ NORTH	
1	R : f0 W : f0		MAC	R : a W : a		R : sA	R : f2 W : f2		MAC	R : c W : c		R : sB	READ NORTH	
2		R : f1 W : f1	MAC		R : b	R : sA W : sA	R : f3 W : f3		MAC		R : d	R : sB W : sB	READ NORTH	
3	R : f0 W : f0		MAC		R : b W : b		R : f2 W : f2		MAC		R : d W : d	R : sB	READ NORTH	
4		R : f1 W : f1	MAC	R : a		R : sA	R : f3 W : f3		MAC	R : c		R : sB		JMP 0