VLSI

PHASE

ONE

Team 7

DESIGN TEAM

PRESENTED BY:

Hossam Ahmed	Sec:1	BN:15
Mohamed Abdallah	Sec:2	BN:17
Mahmoud Mohamed	Sec:2	BN:22
Muhanad Atef	Sec:2	BN:26

MAIN FLOW

1-While LOAD signal is set, Data transmitted from CPU to I/O module through databus in a clock-based manner, Data is decompressed in I/O module, Saved in RAM as shown in RAM block.

Now data is initialized, it's read by Euler provided PROCESS signal (from the coordinator) in the following manner which is organized by an FSM in the data flow control in Euler:-

- N and M, T, precision and solver modes are read together from a slot, then h and e, M and N are used to set N & M registers in Euler. N,M are used to determine the offset in the adress counters in the data flow control unit in Euler which set the address busses reading from the RAM as will be shown, all according to the FSM.
- The main loop then involves reading the next required T entry to set the Next time step register in Euler, then:

o The inner loop involves reading an entry of A and X, an entry of Un and Uz (if U calc yet not calculated for the current time step or U calc entry with B), pass them to the holding registers in Euler and its main functionality, and so on until (Max of (N,M)), a signal (entry done is raised from counter 1 in Euler), that prompts a write signal to the RAM to save an entry in Xn+1 and signify counter2 to increment.

o Back to the inner loop again, all until counter2 raises vector done signal, which signify the end of the current time step and increments the Main counter in Euler.

o in the same time a comparator compares the current Main counter step to the hold reg. that carries T, when they are equal, a Result signal sets which prompts a signal from Euler to the I/O module through the Coordinator (Done signal) to store the answer in the output section in RAM.

-Then back to the main loop again

Note: Un, Uz, X entries are kept read repeatedly with the inner loop, the counters in the data flow control are responsible for setting the needed address bus, according to the address of each section in RAM (with the flow organized by the FSM inside the date flow control)

-

3-Euler's main functionality (first for fixed step with step algorithm module disabled) involves Un, Uz entries to be directed to the interpolator module with the current time step (from Main counter) to the interpolator, getting back Uk entry at that time step, multiply A entry to X entry, add the product to a sum (memory based Full adder), in parallel to multiplying and adding B entry to U entry, until entry done signal is raised from counter1, which enables circuit that calculates Xn+1 entry and replacing the reg. that holds Xn with it and so on.

4-Second, for the variable step algorithm: - an FSM of 4 main states is working in the Coordinator module, its outputs act as selectors for Muxes in order to:

- in state 0, get Xn+1 (0) and store it in a defined section in memory.

- in state 1, updates h reg. to be h/2, gets Xn+1 of the first step

- in state 2, (with h still equals h/2) gets Xn+1 (1), alongside with Xn+1 (0), accumulates the absolute of their differences, use a comparator to compare the error to the tolerance, and decide whether to update h according to the given eqn, and return to state 0 or stuck in state 4 until output is calculated then back to signal (0).

Also, outputs control signals that

- Decrements time step when moving to state 1
- Choose the right addresses of Xn and Xn+1 to read and write in RAM

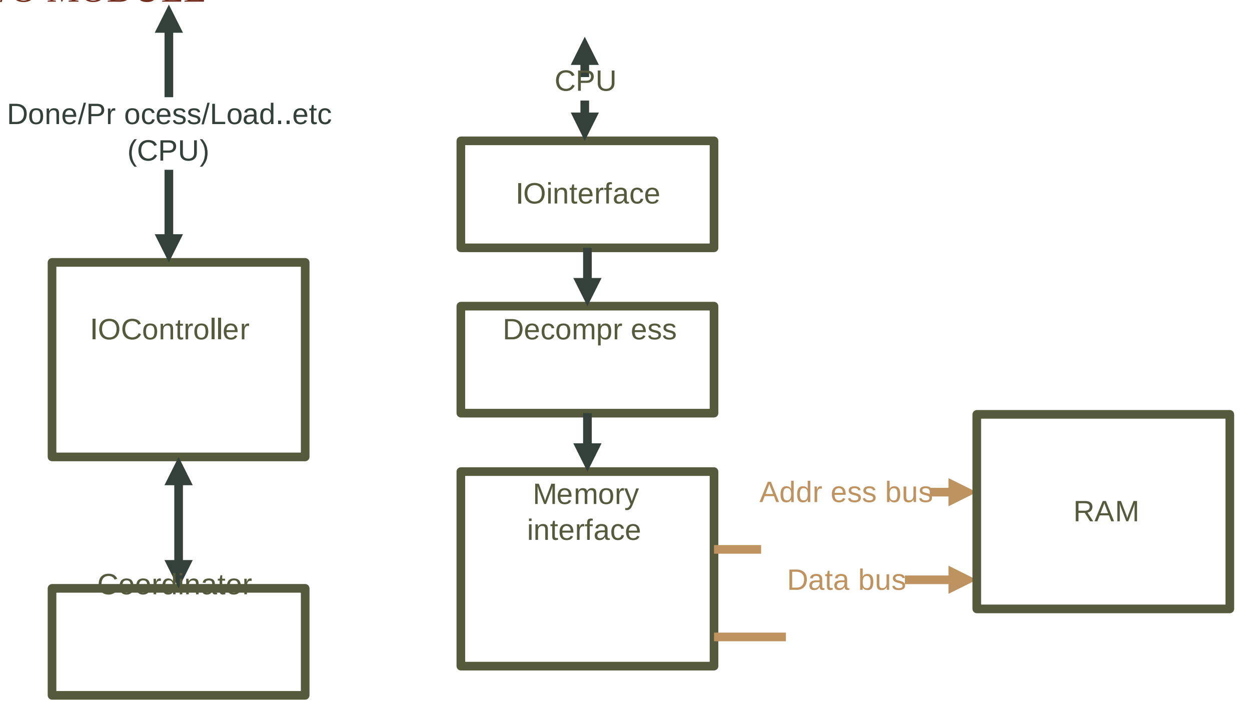
5- When the whole output matrix is done, a signal from Euler prompts output module to start reading the output.

UNITS

Notes:

- We’ve color-coded our figures based on Dark green color is signal Orange color is data.
- Clock and Reset is input in all the shown Modules.
- Interrupt signal is an enable to I/O module.

I/O MODULE



I/O interface:

- Transmit data from CPU to decompression in a clock-based manner.

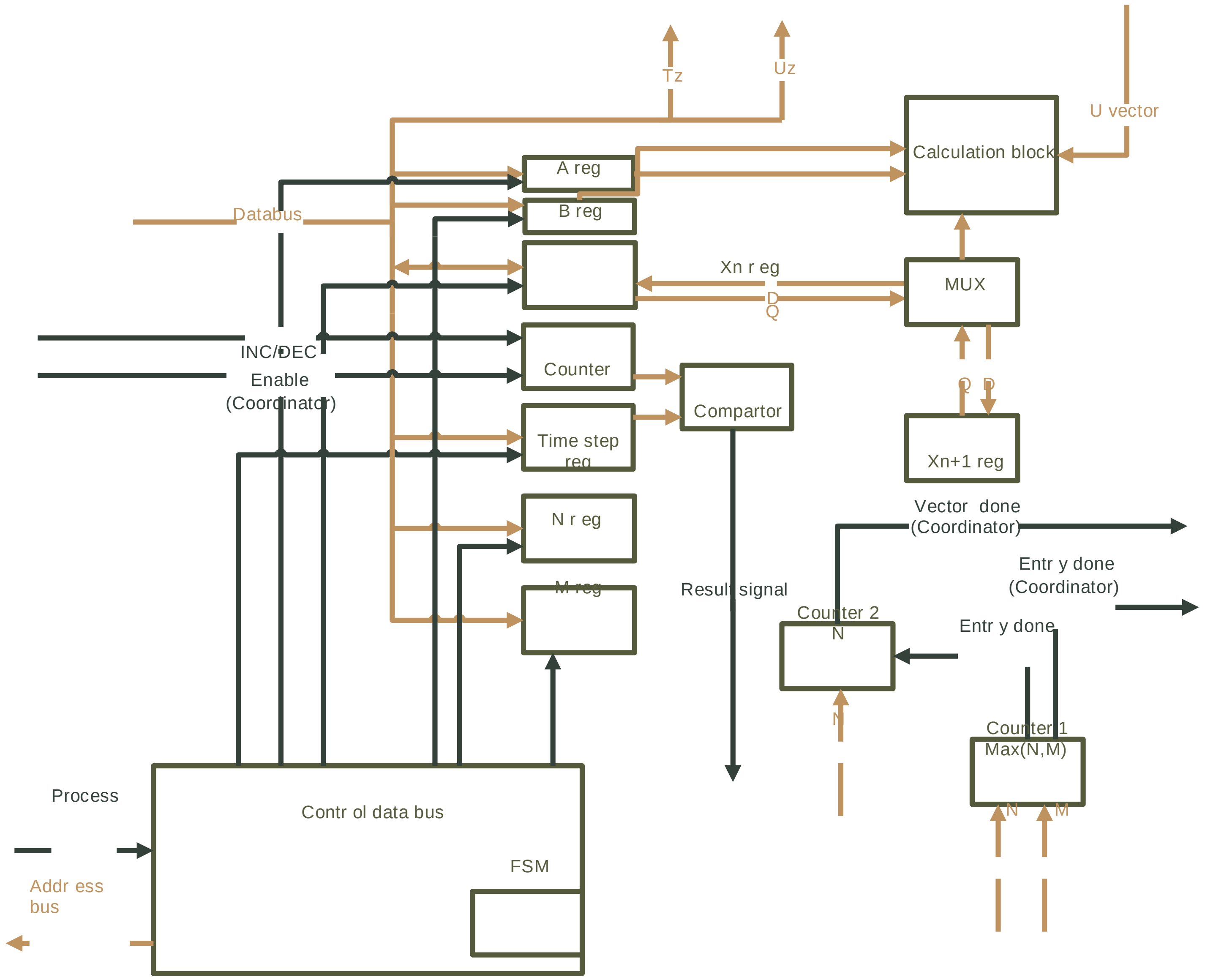
Memory Interface:

- Distributes data at specific slots in RAM.

I/O Controller:

- Send and receive control signals.

FORWARD EULER



Control data bus:

- Controls how the data will be set in the registers using an FSM inside it.
- It will get N then M at first with Process signal.
- Set T
- Set entries of A then set B then X and U.

Note: all these steps are done according to FSM states.

FSM:

- At state0, controls a decoder that outputs "1 0 0 0 0 0" corresponding to IN/OUT control signals N, M, cache registers (A and B), X, U respectively.
- Similarly, it handles data flow to other blocks in the next states.

Comparator:

- Compare the counter with Time points where solutions are required at this time point will raise a "Result" Signal.

Calculation block:

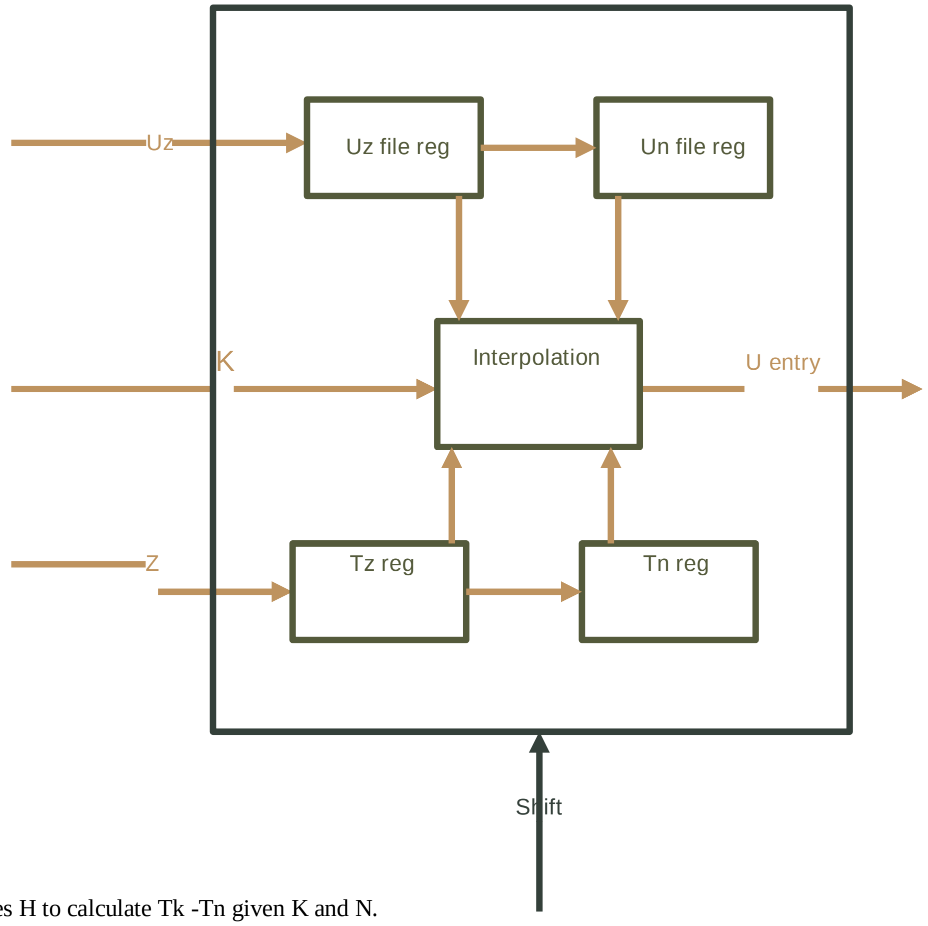
- Compute the main equation of Euler (X_n+1

Counter File:

- Counter one increments until it reaches to the greater of N, M then raise signal "Entry done" to enable counter two to increment .
- Counter two increments until it reaches to N then raise signal "Vector done".
- Both counters reset when reach the maximum,
- Both signals organizes read and write to the RAM.

Edit: a specific counter for each register to read data from its defined section in RAM

INTERPOLATION



Note: it uses H to calculate Tk -Tn given K and N.

Blocks:

- Old time step and U vector (N, U_n).
- New time step and U vector (Z, U_z).
- Main Interpolation Circuit.

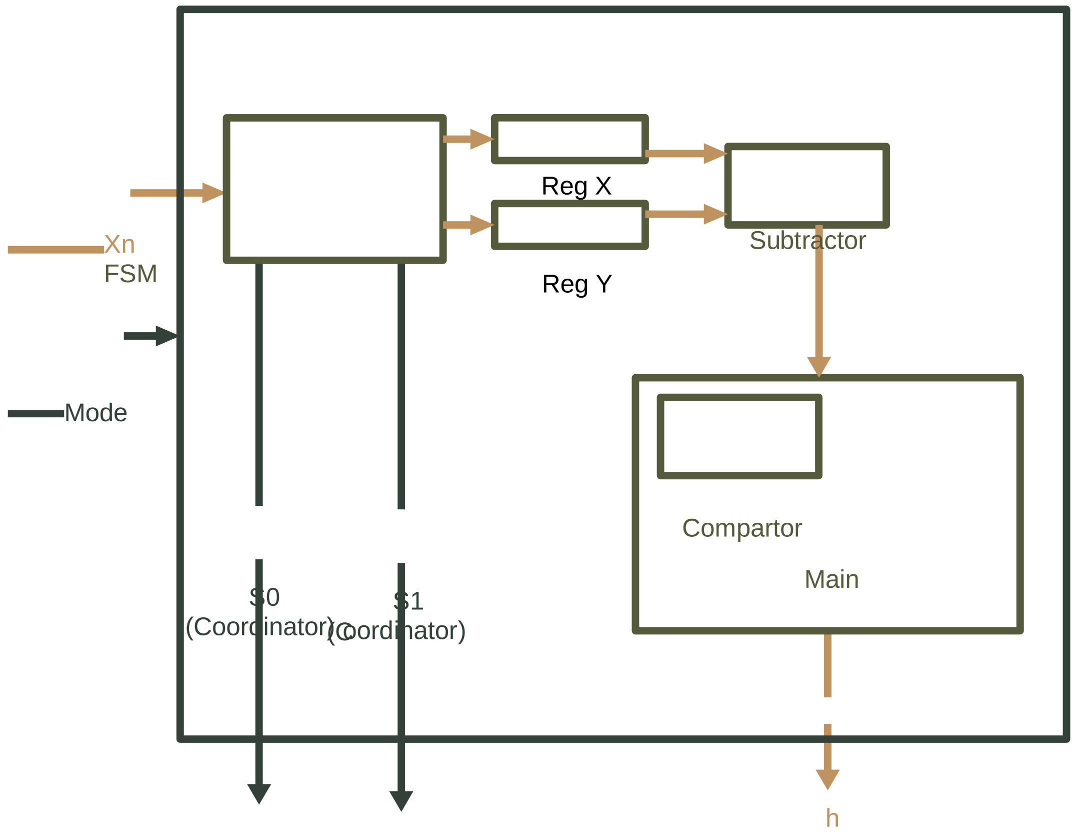
Inputs:

- Given time step (Un, Uz entries) (input from Euler) (every clock cycle)
- Tk every new timestep
- Tz (every output step)
- “Shift” Signal That shifts T_z to T_n and read new T_z. (“Vector done” signal in Euler)

Output:

- U_k (U at given time step) (output to Euler)

STEP ALGORITHM



FSM:

- state0: get Xn+1 (0) from Euler and store it in regX.
- state2: updates h reg. to be h/2.
- state3: (with h still equals h/2) gets Xn+1 (1) from Euler, store it in regY.
- State (Two bits) which describe state to Coordinator to raise INC/DEC and Enable signals to the main counter and select X_n or X_{n+1}.

Subtractor:

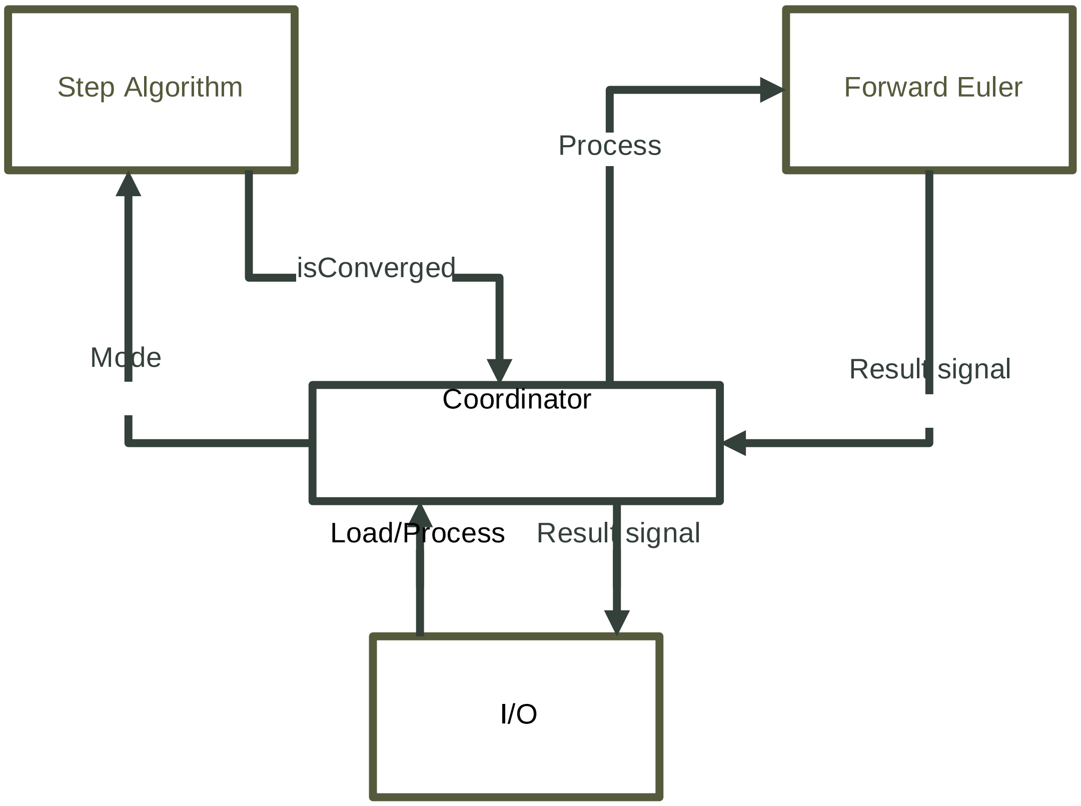
- Calculates the difference between $X_{(n+1)}^{1}$ - $X_{(n+1)}^{0}$

Main:

- It loops until error converges and it raises a signal “Converge” to the coordinator., which is then conveyed to Euler to be used by the data flow control to decide whether to increment address of the read X entry, or read the same one.

Edit: the FSM is the coordinator's responsibility, step module only calculates Hnew according to the given equation.

COORDINATOR

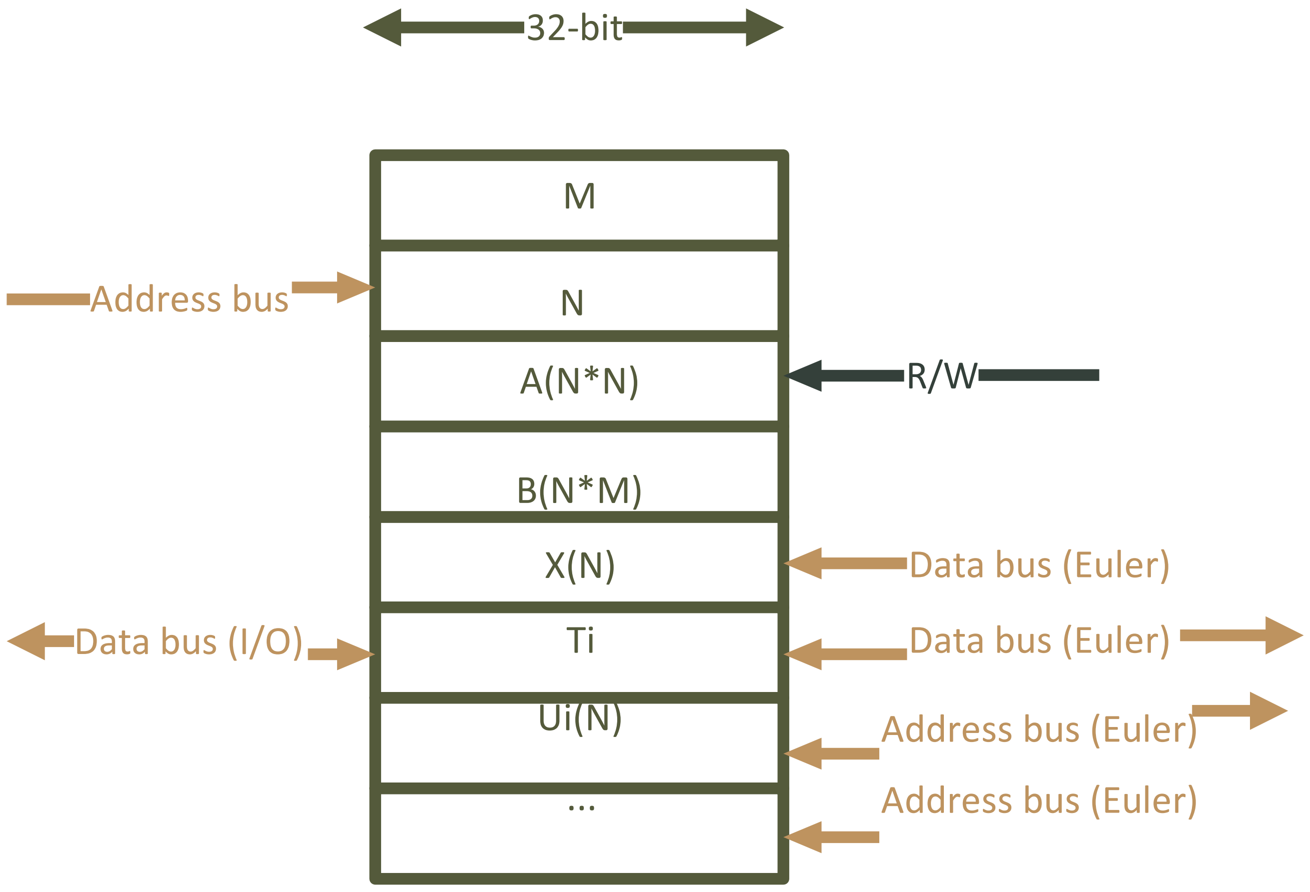


Flow:

1. Convey load/process signal from I/O to Euler, Mode signal to Step Algorithm.
2. Transmit converge signal from Step Algorithm to Euler to enable counter to increment.
3. Receive result signal from Euler and prompt I/O to read result from RAM.

Edit: Coordinator is responsible for the orchestrating FSMs of the databus control and variable step algorithm.

RAM



Note: L and h are kept bellow M and N

RAM Edit:

For the sake of latency, we'd better read 4 values from RAM (A, X, B, U) every clock cycle (through 4 ports), instead of reading them in 2 consecutive cycles, that's why RAM is now divided into 2 memories with 2 ports each (one Read only and the other is bi-directional).

RAM organization:

Memory1:

N, M, T, precision mode, solver mode
e
h
A[50 * 50]
X[50 * 5]
Ti [5]
XS1[50]
XS2[50]

Note: X block carries X's at the required output times, XS1, XS2 are the temporary storage needed by variable step algorithm to calculate final h.

Memory2:

B[50 * 50]
Ucalc [50]
U [50 * 5]

Note: U is the full matrix, Ucalc is the interpolated U vector representing the current time step (overwritten each time step).

Name	Task	Hours spent On task **(Hours)**	Problems Faced
All Team	Overall Scenario	4	Link between Fixed and variable step Algorithms.
	I/O Module	0.5	-
	Forward Euler	3.5	Databus control and interaction with step Algorithm.
	Step Algorithm	2.5	Module Responsibility wasn't clear enough.
	Interpolation	1.5	Translating Logic of Interpolation into Hardware.
	Coordinator	1	-
	RAM	1	RAM Specs. And word size.