COMPUTER ARCHITECTURE: ASSIGNMENT 1

DEPT: CSE (A)
ROLL NO: 1651009

1. Explain masking and data routing mechanism in an array processor (i.e. SIMD)

Ans: A synchronous array of parallel processors is called an array processor which consists of multiple processing elements (P.E.s) under the supervision of one control unit (C.U). The components of a P.E. is illustrated in the following figure:

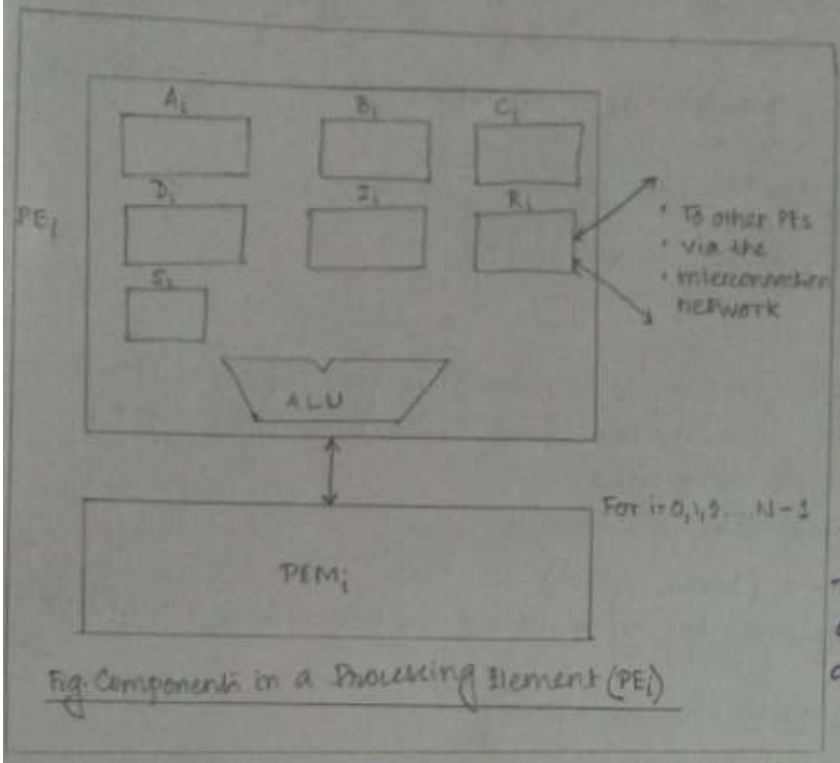

Fig. Components in a Processing Element (PE_i)

In the adjacent figure, we see, each P.E. is a processor with its own memory PEM_i, a set of working registers and flags; namely A, B, C and S; an ALU; a local index register I_i; an address register D_i and a data routing register R.

The R_i of each PE_i is connected to the R_i of other PE_i via the interconnection network. When the data transfer among PEs occur the contents of R_i registers are being transferred.

Some array processors use 2 routing registers — for i/p and o/p but we have considered the use of 1 routing register (R_i) per PE_i in which the inputs and outputs of R_i are totally isolated by using master-slave flipflops.

PE is either in active or inactive mode during instruction cycle
If a PE is active, it executes the instruction broadcast to it by the CU, otherwise it will not

Masking schemes are used to specify the status flag S_i of PE_i
$S=1$ indicate active PE_i and $S=0$ for inactive PE_i.
In CU there is a global index register I and a masking register M. The M register has N bits. The ith bit of M is denoted as M_i.
The collection of S_i flags for $i = 0, 1, 2, \ldots, N-1$ forms a status register S for all the PEs.
Note that the bit patterns in registers M and S are exchangeable upon the control of the CU when masking is to be set.

2. Explain the speedup in case of the following instruction level pipeline architectures: linear pipeline, superscalar pipeline and superpipeline architectures?

Ans: • Linear pipeline

Ideally, a linear pipeline of k stages can process n tasks in k+(n-1) clock cycles, where k cycles are needed to complete the execution of the very first task and the remaining (n-1) tasks require (n-1) cycles. Thus the total time required is

$$T_k = [k+(n-1)]\tau$$

where τ is the clock period. Consider an equivalent-function non-pipelined processor which has a flow-through delay of $k\tau$. The amount of time it takes to execute n tasks on this nonpipelined processor is $T_1 = nk\tau$.

The speedup factor of a k-stage pipeline over an equivalent non-pipelined processor is defined as

$$S_k = \frac{T_1}{T_k} = \frac{nk\tau}{k\tau+(n-1)\tau} = \frac{nk}{k+(n-1)}$$

• Superscalar Pipeline

To compare the relative performance of a superscalar processor with that of a scalar base machine, we estimate the ideal execution time of N independent instructions through the pipeline.

The time required by the scalar base machine is

$$T(1,1) = k+N-1 \text{ (base cycles)}$$

The ideal execution time required by an m-issue superscalar machine is

$$T(m,1) = k + \frac{N-m}{m} \text{ (base cycles)}$$

where k is the time required to execute the first m instructions through the m pipelines simultaneously, and the second term corresponds to the time required to execute the remaining (N-m) instructions, m per cycle, through m pipelines.

The ideal speedup of the superscalar machine over the base machine is

$$S(m,1) = \frac{T(1,1)}{T(m,1)} = \frac{N+k-1}{N/m+k-1} = \frac{m(N+k-1)}{N+m(k-1)}$$

As $N \to \infty$, the speedup limit $S(m,1) \to m$, as expected.

• Superpipeline

The minimum time required to execute N instructions for a superpipelined machine of degree n with k stages in the pipeline is

$$T(1,n) = k + \frac{1}{n}(N-1) \text{ (base cycles)}$$

Thus the potential speedup of a superpipelined machine over the base machine is

$$S(1,n) = \frac{T(1,1)}{T(1,n)} = \frac{k+N-1}{k+(N-1)/n} = \frac{n(k+N-1)}{nk+N-1}$$

The speedup $S(1,n) \to n$, as $N \to \infty$.