

# Datapath Scheduling using Dynamic Frequency Clocking

**Saraju P. Mohanty, N. Ranganathan**

Dept. of CSE, University of South Florida,  
Tampa, FL 33620, USA

[smohanty, ranganat@csee.usf.edu](mailto:smohanty, ranganat@csee.usf.edu)

**V. Krishna**

Agilent Technology, Palo Alto, CA 94303

[vamsi@labs.agilent.com](mailto:vamsi@labs.agilent.com)



# Outline of the Talk

- Related work
- Why frequency variations ?
- What is dynamic frequency clocking (DFC) ?
- How DFC can save energy & improve performance ?
- Target Architecture
- Scheduling Algorithm (DFCS)
- Experimental results
- Conclusions

# Related Work

## Low Power Scheduling using Voltage Reduction

- Chandrakasan, et al.[4], 1995 - transformations
- Johnson & Roy [8], 1997 – Mover algorithm, multiple voltage
- Chang & Pedram [10], 1997 – Dynamic programming
- Raghunathan & Jha [11], 1997 – SCALP
- Kumar & Bayoumi [15], 1999 – variable voltage
- Sarrafzadeh and Raje [17], 1999 - dynamic prog., geometric
- Schiue & Chakrabarti [18], 2000 – list based, multiple voltage
- Manzak & Chakrabarti, IEEE Trans. VLSI Systems, Vol.10, No.1, pp. 6-14, Feb 2002 - energy minimization using Lagrange multiplier method
- And many other works

## Related Work: MOVER [8]

- ILP formulations of multiple supply voltage scheduling (MOVER)
- Minimize energy consumption
- The DFG is partitioned into groups :
  - Higher voltage operation group
  - Lower voltage operation groupMOVER 1<sup>st</sup> fixes the minimum voltage of the lower group and then fixes the minimum voltage for the upper group
- Handle timing and resource constraints
- Exponential worst-case complexity
- Energy Savings : 0-50%, Area Penalty: 0-170%

## **Related Work :** Dynamic Programming[10]

- Dynamic Programming for multiple voltage scheduling
- Time-constraints scheduling
- Average energy savings : 40 %
- Non-exponential complexity
- Can handle large circuits, pipeline datapath

## Related Work : List-Based [18]

- Multiple voltage scheme
- List based scheduling algorithms: resource constraints based and time constraints based
- Polynomial time complexity algorithms
- Multiple voltage scheduling and reduce switching activity
- Savings : Latency constrained case : 13-77%

# Why Frequency Variations?

1. Energy dissipation per operation,  $E = C_{eff} * V_{dd}^2$
2. Power dissipation per operation,  $P = C_{eff} * V_{dd}^2 * f$
3. Delay that determines maximum frequency,  
 $t_d = k * V_{dd} / (V_{dd} - V_T)^\alpha$

where,  $\alpha$  is a technology dependent factor,  $k$  is a constant and  $1/t_d = f_{max}$

## What we deduce from the equations ?

1. If we reduce supply voltage ( $V_{dd}$ ), delay increases and performance degrades.
2. If we reduce clock frequency ( $f$ ), we save only power, but do not save energy.
3. If we reduce switching activity, we reduce the effect of ( $C_{eff}$ ) as well as correlations to an extent. – this needs to be done irrespective of 1 and 2 above.

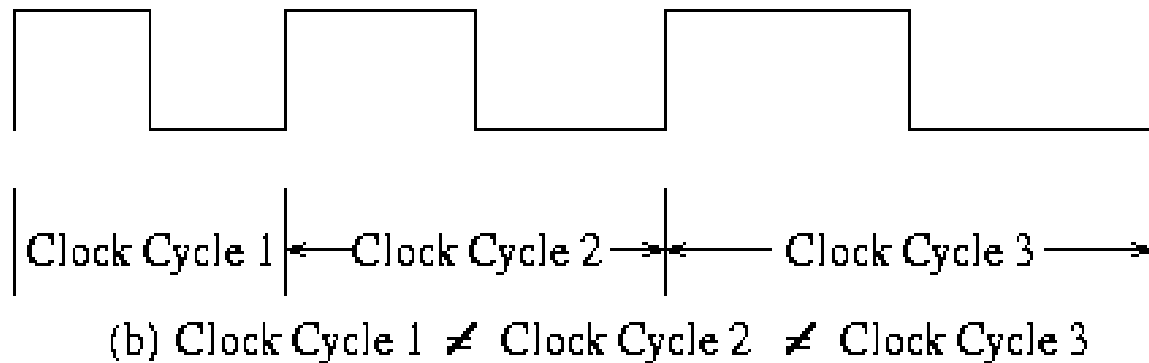
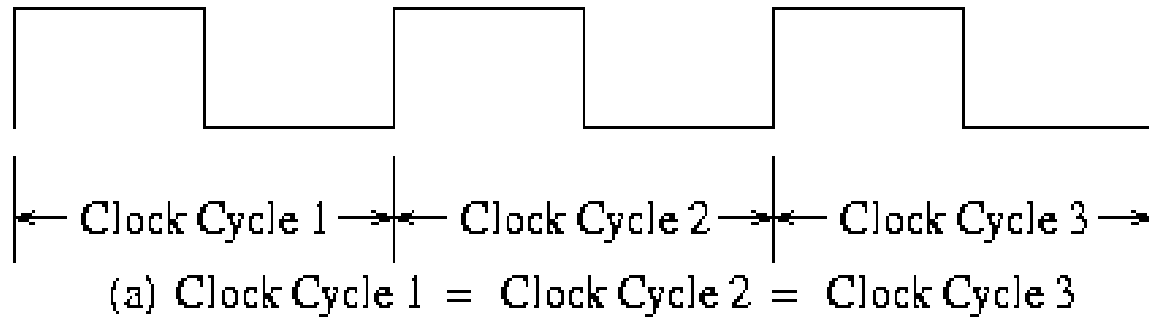


# Our Approach

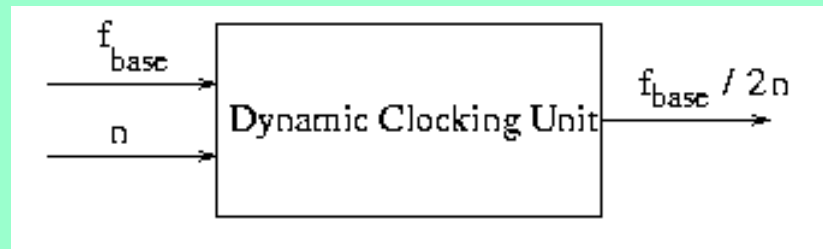
Adjust the processor's frequency and reduce the supply voltage together during scheduling

By doing that we reduce energy while maintaining performance or even improve performance if possible!

# Dynamic Frequency Clocking (DFC)



# Dynamic Clocking Unit (DCU)

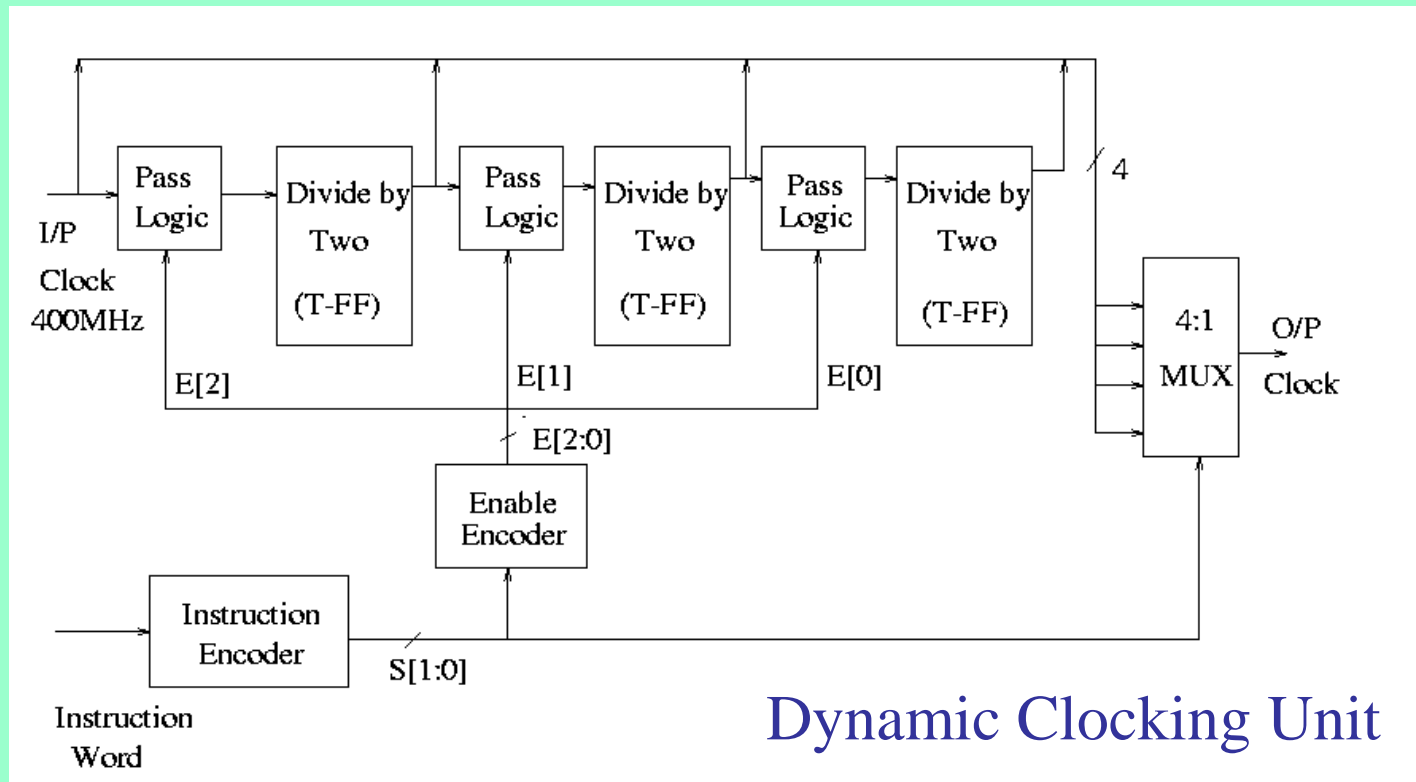


DCU uses clock divider strategy.

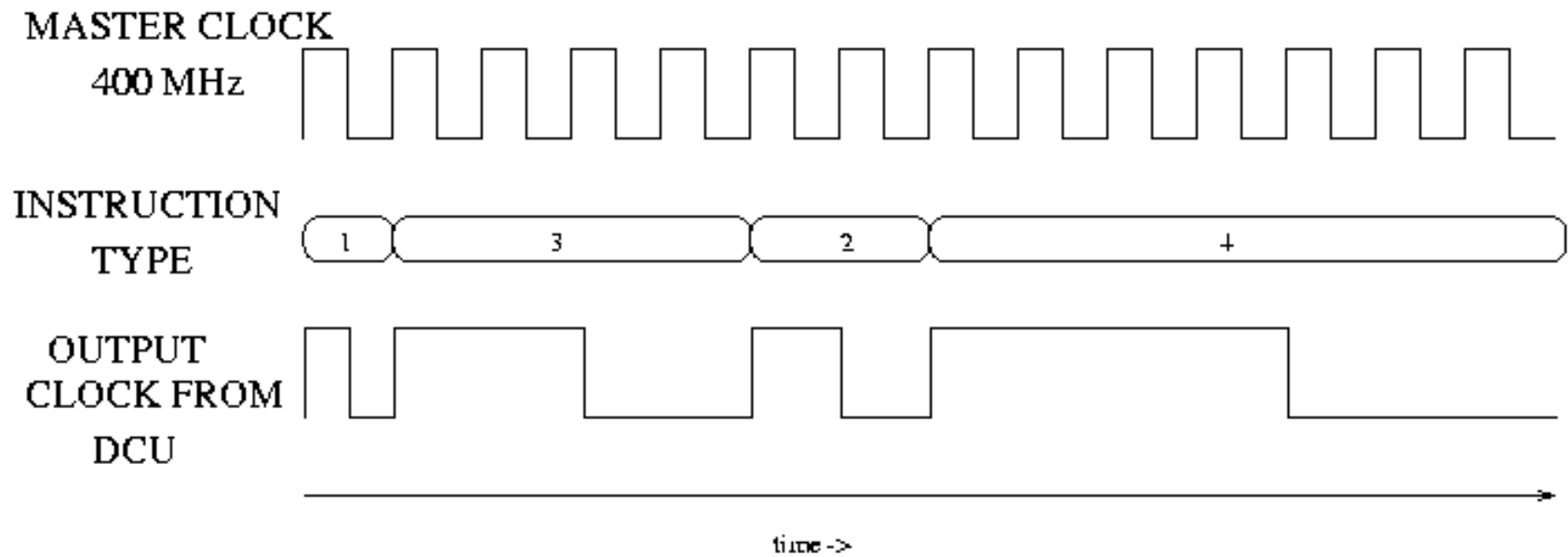
More details : Ranganathan [5, 12]

# DCU Details : DFLAP Architecture[5, 12]

- Linear array processor (DFLAP) designed using Dynamic Frequency clocking
- The chip is operated at different frequencies switching dynamically depending on the instruction being executed



# DCU Details : DFLAP [5] .....

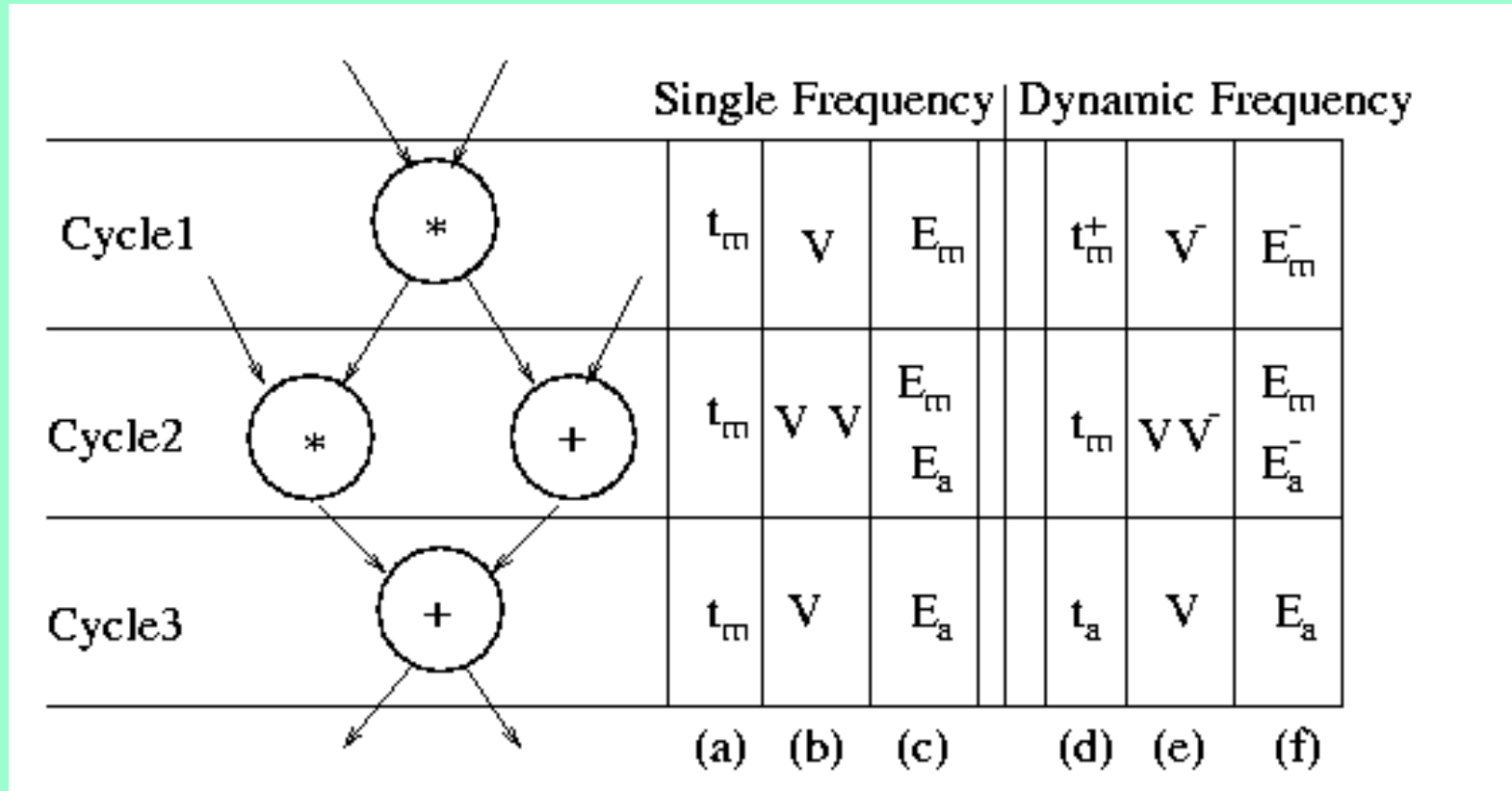


INSTRUCTION TYPE : FREQUENCY OF OPERATION

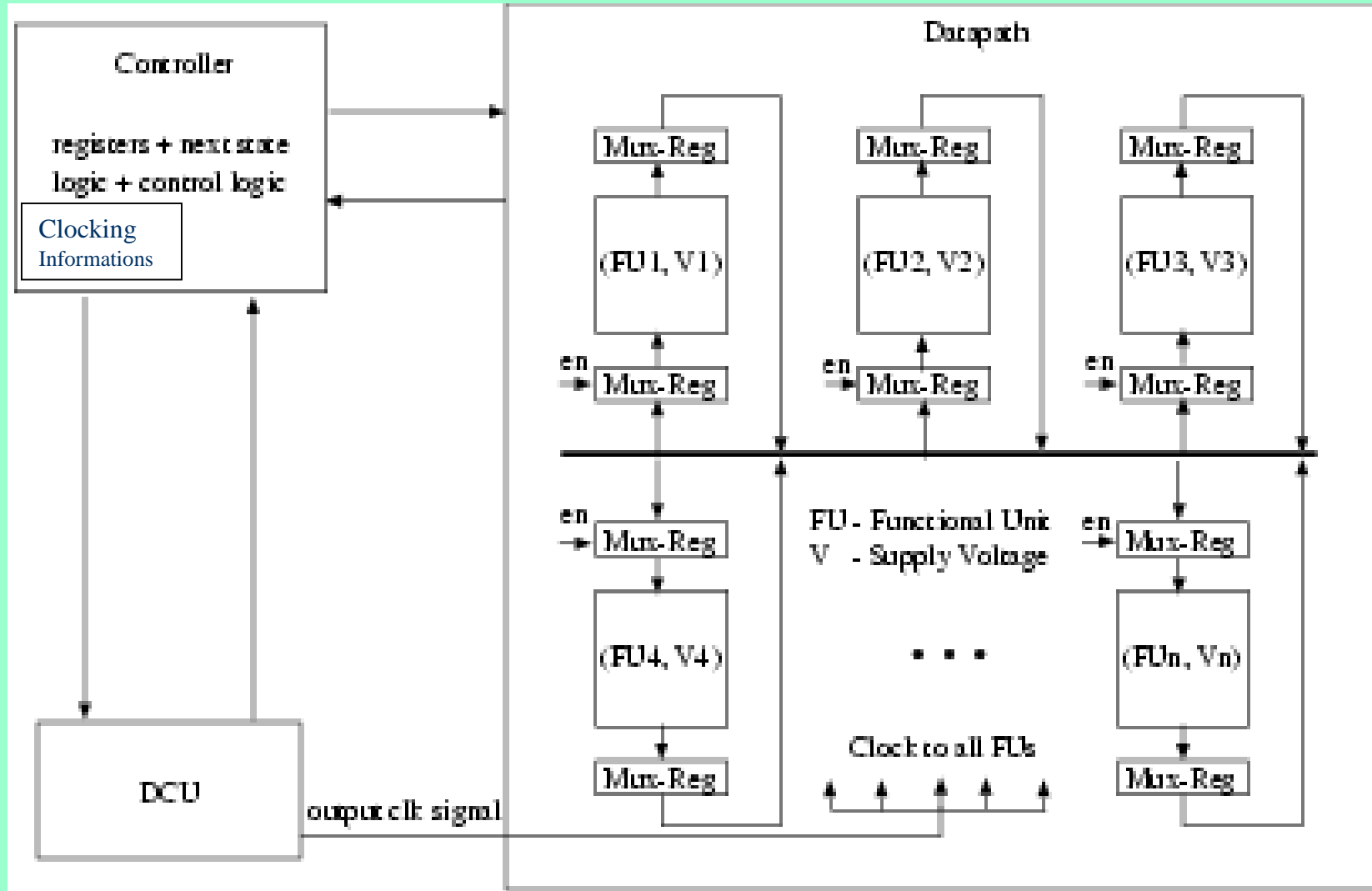
1	400MHz
2	200MHz
3	100MHz
4	50MHz

Waveform generated by DCU

# DFC for energy savings and performance



# Target Architecture



# Operating Frequencies

<b>Components .8 micron 16-bit cells</b>	<b>Delay / Freq. (5V)</b>	<b>Delay / Freq. (3.3V)</b>	<b>Delay / Freq. (2.4V)</b>
Adders/ Subtract (ALU)	35.0ns	62.2ns	105.3ns
Max. frequency	28.5MHz	16.07MHz	9.49MHz
Scaled down frequency	28 MHz	14 MHz	7 MHz
Multipliers (MULT)	63.3ns	113.3ns	192.2ns
Max. frequency	15.8MHz	8.82MHz	5.2MHz
Scaled down frequency	14 MHz	7 MHz	4.6 MHz



# DFCS Scheduling Algorithm

What it does?

**Attempts to operate high energy units (multipliers) at lower frequencies and low energy units (ALUs) at higher frequencies to save energy without losing performance as much as possible**

# DFCS Algorithm

**Input:**        Unscheduled DFG, time constraint (execution time for critical path), operating frequencies and voltages

**Output:**     Scheduled DFG with frequency assignment for each control step (cycle) and voltage assignment for each vertex (operation)

# DFCS Algorithm

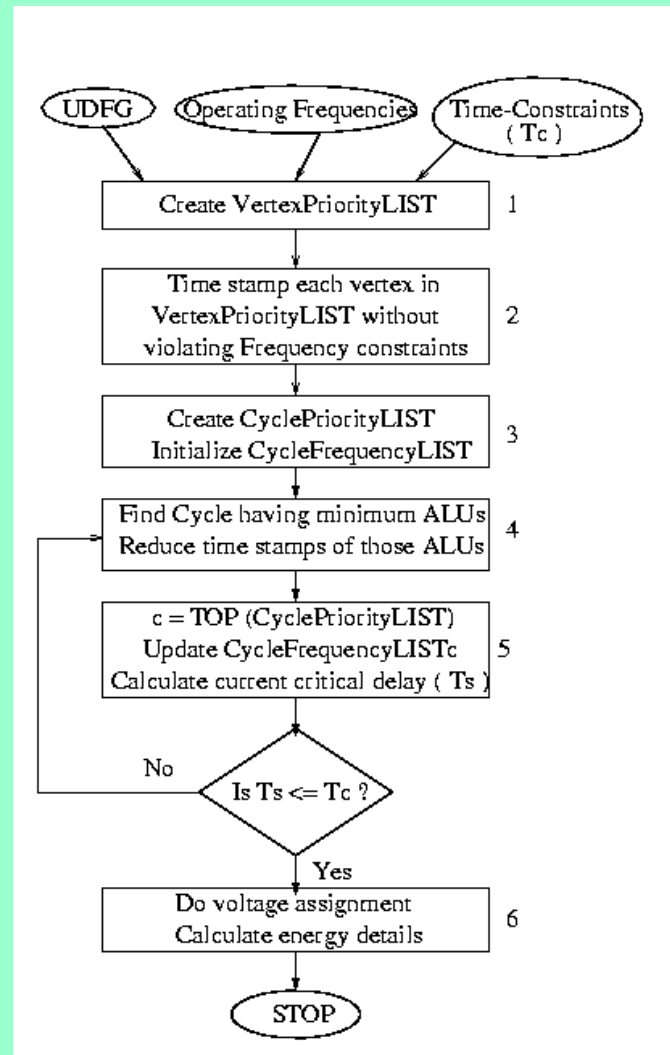
**Vertex priority list:** the vertices from source to sink in DFG are reordered such that the multipliers are grouped with higher priority than adders and among the multipliers (and adders) the precedence in DFG is maintained. (used to ensure precedence by time-stamping)

**Cycle priority list:** the control steps or the cycles are reordered in this list such that the cycles with more simultaneous operations (consuming more energy) get higher priority. (used for low frequency assignment in the algorithm)

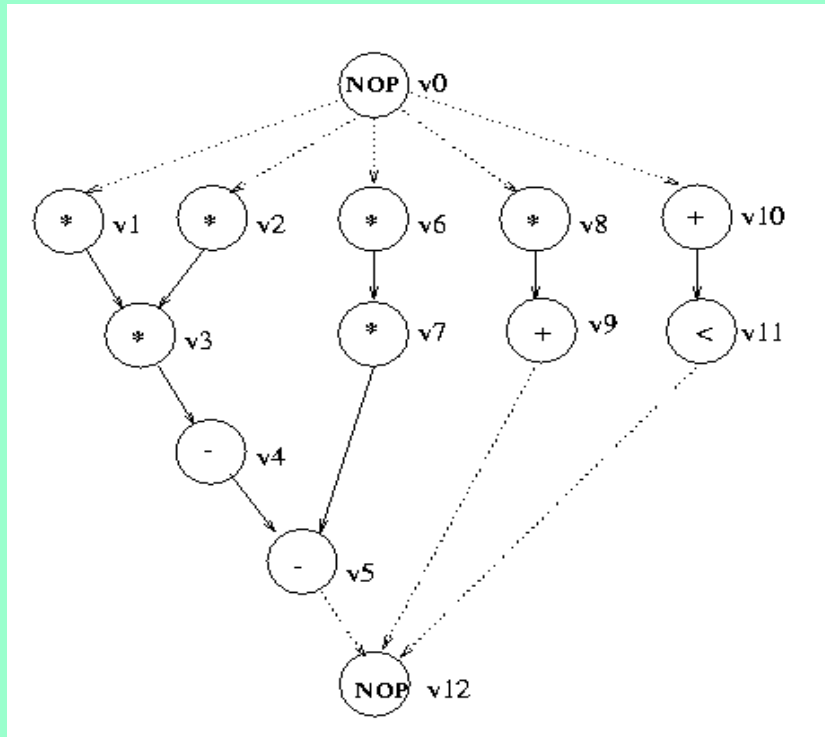
# DFCS: Algorithm

- Step 1: create vertex priority list of vertices in DFG
- Step 2: assign control steps to the operators such that precedence is satisfied and that multiply and add operators are not scheduled in the same cycle - yields an intermediate schedule
- Step 3: create a cycle priority list using the intermediate schedule
- Step 4: assign frequency to each control step using cycle priority list – assign higher frequency to lower priority steps
- Step 5: If execution time not close to time constraint, then eliminate the control step that has minimal number of ALU operations also adjusting all its predecessors to get a new intermediate schedule. Go to Step 3.
- Step 6: To the schedule with frequency assignment for each control step, perform voltage assignment for each operation

# DFCS Flow Chart

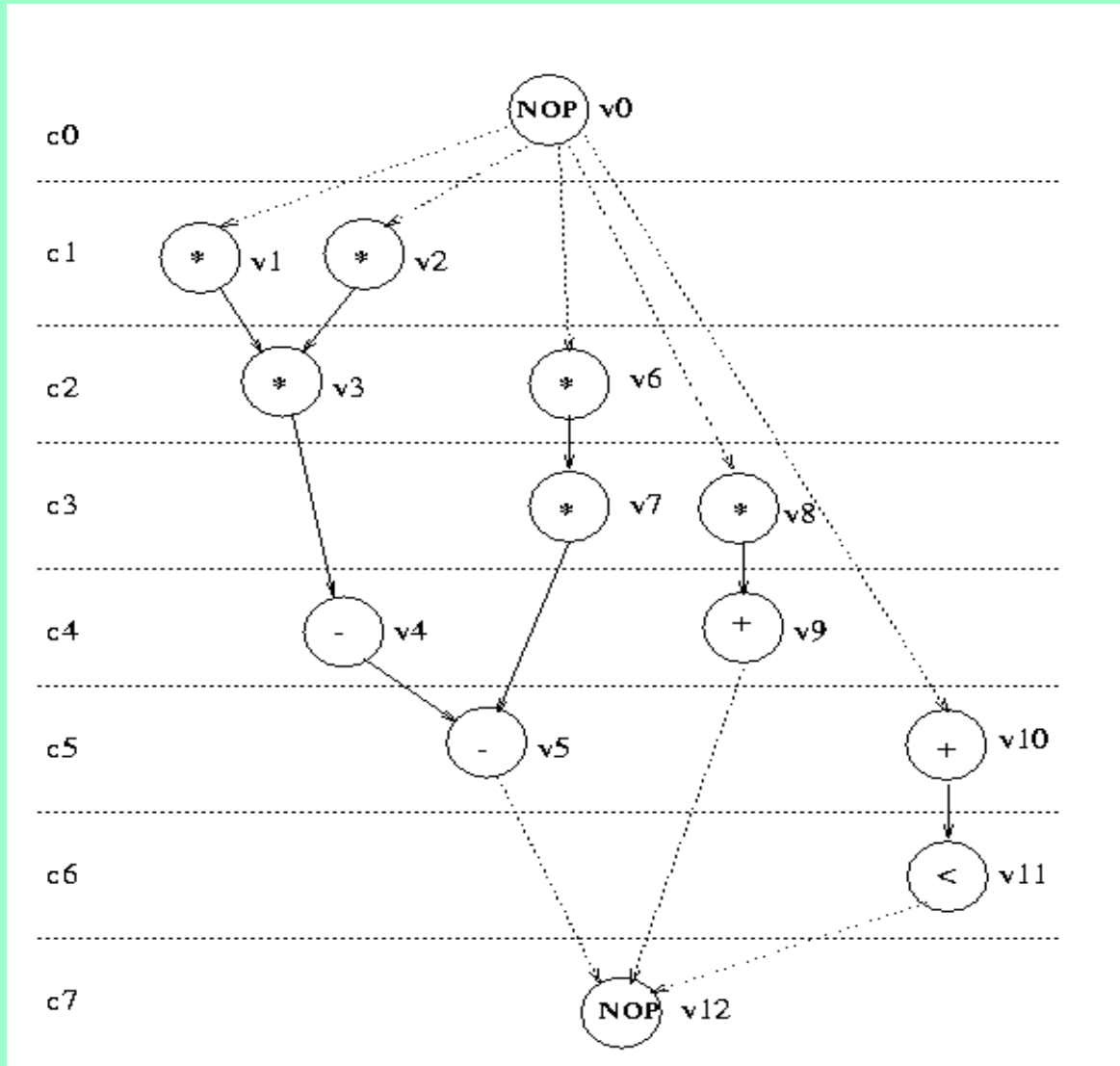


## Example to illustrate DFCS



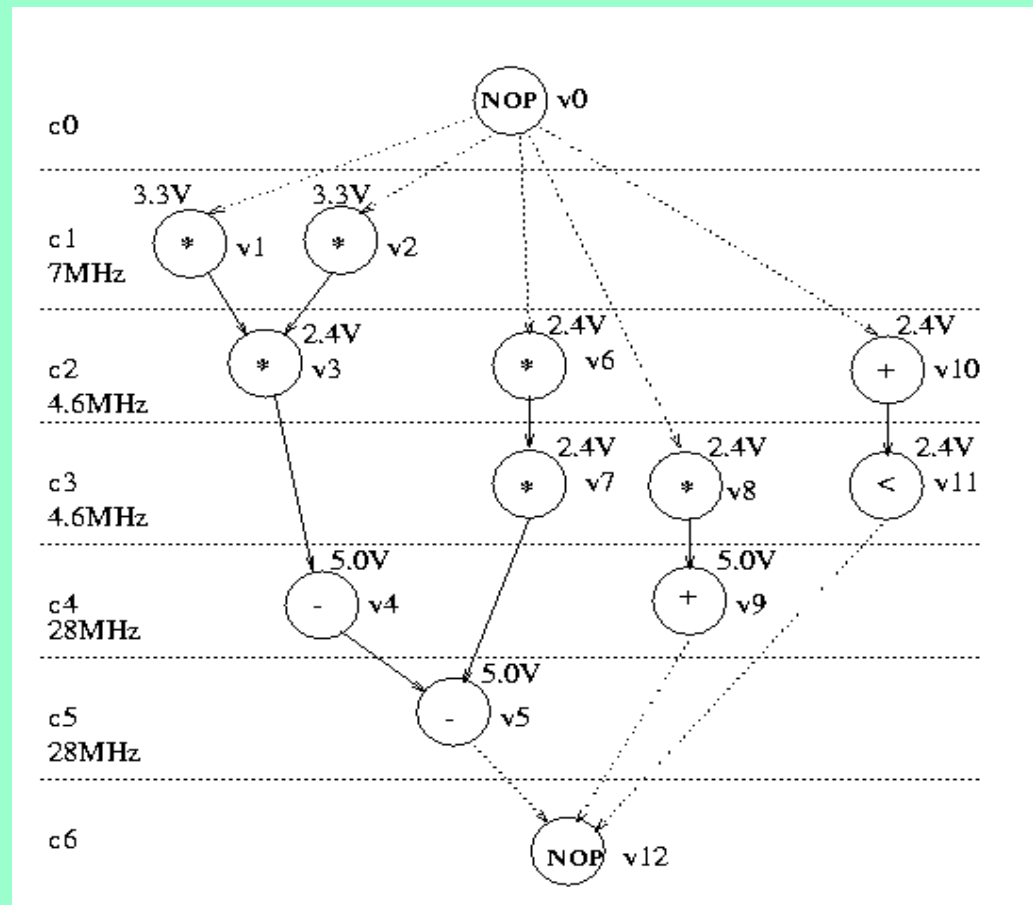
v0	v1	v2	v3	v6	v7	v8	v4	v5	v9	v10	v11	v12	Vertices
0	1	2	3	4	5	6	7	8	9	10	11	12	Priorities

# Initial Intermediate Schedule



## Final Schedule for Example

c0	c2	c3	c1	c4	c5	c6
----	----	----	----	----	----	----

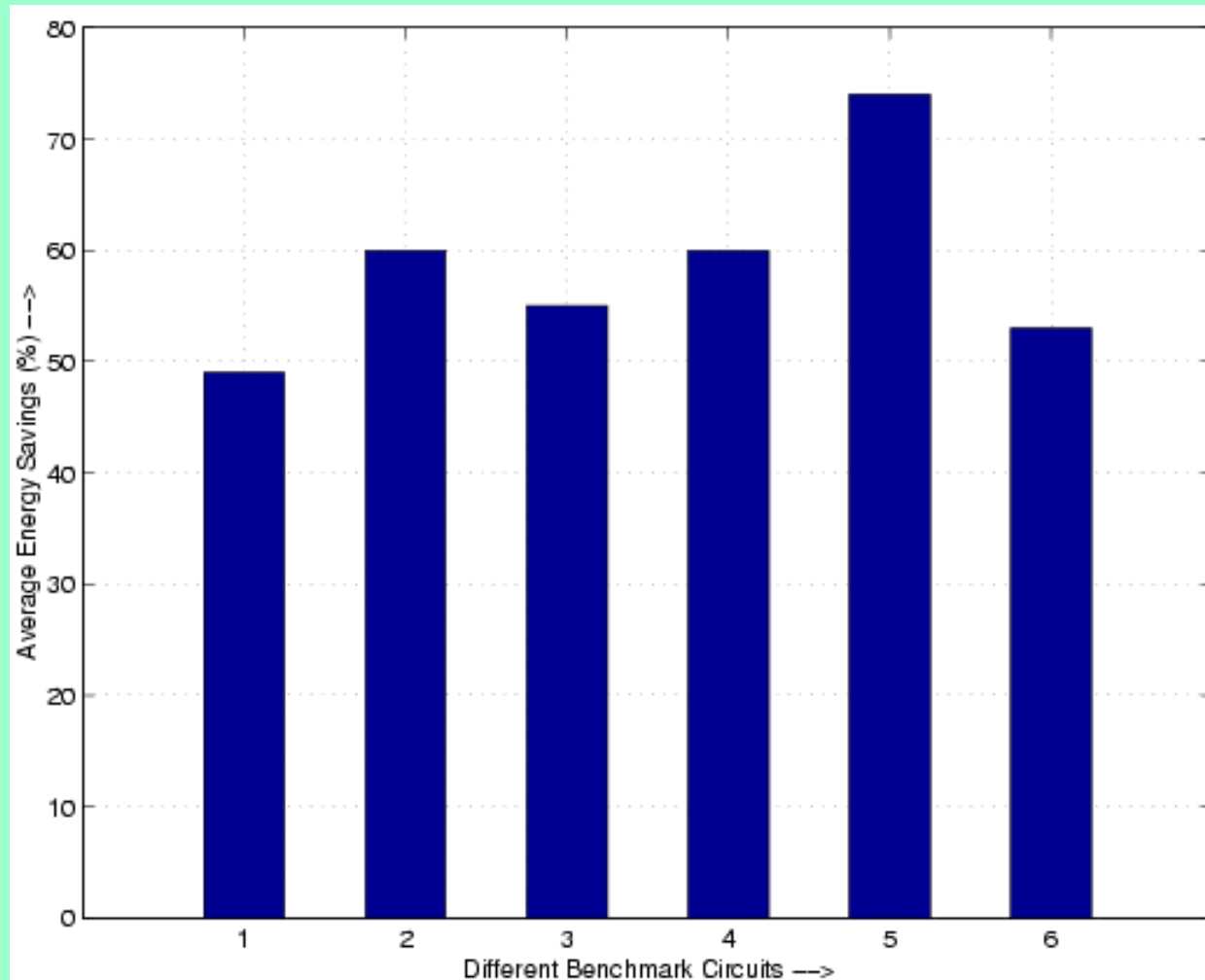




## Savings for different benchmarks

Benchmarks	Time Constraints (ns)	$E_{single}$ (pJ)	$E_{dynamic}$ (pJ)	% Savings
1. AR	$1.5T_{cp}$ , $1.75T_{cp}$ , $2.0T_{cp}$	36186	21491, 18139, 15274	40.61, 46.61, 57.79
2. BPF	$1.5T_{cp}$ , $1.75T_{cp}$ , $2.0T_{cp}$	27672	15187, 9350, 8249	45.12, 66.12, 70.19
3. EWF	$1.5T_{cp}$ , $1.75T_{cp}$ , $2.0T_{cp}$	19422	12335, 8814, 5341	36.49, 54.62, 72.50
4. FDCT	$1.5T_{cp}$ , $1.75T_{cp}$ , $2.0T_{cp}$	30675	14611, 14489, 7714	52.37, 52.77, 74.85
5. FIR	$1.5T_{cp}$ , $1.75T_{cp}$ , $2.0T_{cp}$	18696	4910, 4877, 4820	73.74, 73.91, 74.21
6. HAL	$1.5T_{cp}$ , $1.75T_{cp}$ , $2.0T_{cp}$	13614	7808, 6821, 4449	42.64, 49.90, 67.31

## Average savings for 3 time constraints



## Energy savings using different algorithms

Benchmarks	DFCS	Chang [10]	Sarrafczadeh [17]	Johnson [8]	Shiue [18]	Krishna [16]
AR	41-58	40-63	16-20	16-59	38-76	3-53
BPF	45-70	-	-	-	-	-
EWf	36-73	44-69	13-32	11-50	13-76	53-54
FDCT	52-75	43-69	-	-	-	-
FIR	74-74	-	16-29	28-73	-	53-53
HAL	43-67	41-61	-	-	22-77	-

# Conclusions

- A time-constrained scheduling algorithm discussed
- In DFC, frequency can be switched dynamically
- Lowering frequency with voltage can save energy
- For DFCS average energy savings is 46% to 68%
- As the time-constraint is relaxed the energy savings increases
- For the circuits having almost equal number of addition and multiplier operations savings is maximum

Thank you