Instructions-Wake-Up Mechanisms: Power and Timing Evaluation

Marco A. Ramírez[1,4], Adrian Cristal[1], Alexander V. Veidenbaum [2], Luis Villa, [3] Mateo Valero [1]

[1] Computer Architecture Department U.P.C. Spain
+34-93-401-69-79, Fax: +34-93-401-70-55
e-mails: {mramirez, adrian, mateo}@ac.upc.es,
[2] University of California Irvine CA
alexv@ics.uci.edu
[3] Mexican Petroleum Institute, Mexico.
lvilla@imp.mx
[4] National Polytechnic Institute, Mexico

Abstract. Power dissipation is a major constraint in the design of new micro-architectures for state-of-the-art microprocessors. One of units that have received considerable attention as a major consumer of power is the instruction queue. This research studies the power consumption and timing of two different designs of the instruction queue and wakeup logic used in modern microprocessors. One design is a standard CAM-based mechanism; the other is a new proposal using a RAM-based direct wakeup mechanism [7]. The energy and timing of CAM and RAM structures used in the instruction queue were evaluated using Spice3 tools for the 70nm technology.

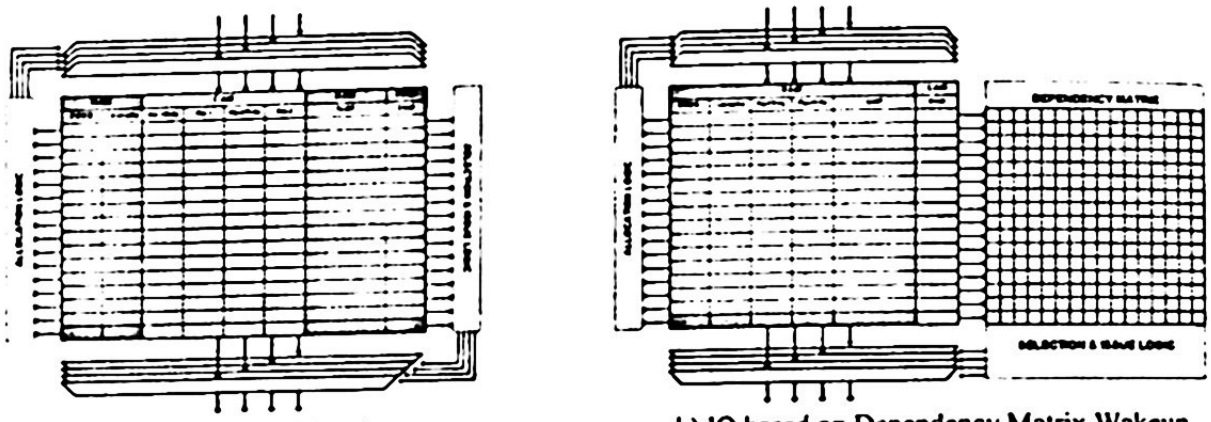
Keywords: Processors Architecture, Issue Queue, CAM Wakeup, Matrix Dependency Wakeup, Direct-Wakeup, Low-Power.

1 Introduction

This work evaluates power and timing of structures which implement the instruction wakeup mechanisms used in modern superscalar processors. It compares traditional designs with a Direct Wakeup scheme proposed in [7]. We pay close attention to the amount of hardware that must be added in order to achieve a power-efficient solution. Section 2 describe how the structures work. Section 3 presents the technology considerations taken into account for evaluations. Section 4 shows a transistor level design of structures used, section 5 presents a new design we propose and its evaluation. Finally, section 6 presents the results and conclusions.

2 Traditional Issue Queues

Instruction queues are structures that permit Out-Of-Order issue and execution, necessary to achieve high Instruction Level Parallelism **ILP** required in modern microprocessors. One operation that has to be performed in a single cycle by the instruction queue logic is Wakeup-Select.



a) IQ based on CAM wakeup

b) IQ based on Dependency Matrix Wakeup

Fig. 1. Issue queue Models

The wakeup logic is responsible for waking up instructions waiting in the issue queue for their source operands to become available. This wakeup action is performed by a Content Addressable Memory. A traditional instruction queue dissipates between 18 and 25% of the total processor power dissipation [5]. A major part of this energy dissipation is due to comparisons performed in the CAM's by the wakeup logic.

There are two ways to wake up instructions (CAM wakeup and Dependency Matrix Wakeup). The first one is by comparing the result tag of a completing instruction with the tags of all source operands of instructions waiting in the issue queue. On a match a respective OpRdy bit is set. Instructions with all its operands available are ready to issue in next cycle. The second way is to perform the comparisons in allocation stage [4]. In this case the source operands are compared with all destination register waiting in the issue queue; matches are used for set the corresponding bits in a dependency matrix. Bits set in a row of the matrix correspond to the instructions to wakeup. Dependency matrix uses column bits counters to evaluate ready instruction, but not make comparisons after the execution. In both two cases, a CAM is required. But the number of comparison ports in CAM-Wakeup model is proportional to the execution width (EW), while in Matrix-Dependency-Wakeup model it is proportional to the decode width (DW).

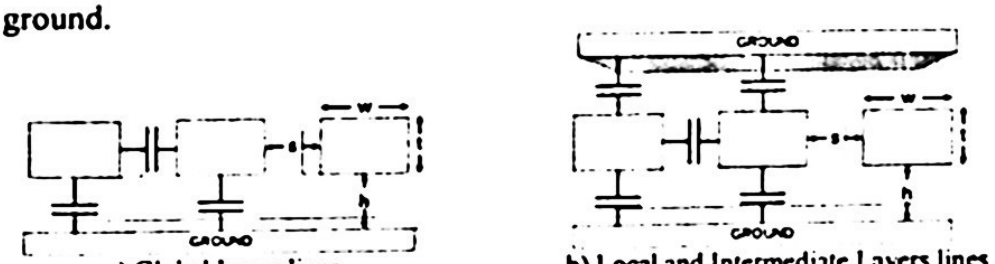
Selection logic determines which instruction has its operands ready; instructions with all operand available are ready to being issue.

3 Technology Considerations

Traditionally, fanout-of-four (FO4) delay is a metric used to estimate CMOS-circuit speed because it is independent of technology process. The FO4 delay is the time for an inverter to drive 4 copies of same inverter. For static logic, an approximation of the FO4 delay in picoseconds is given by $360 \times L_{drawn(\mu m)}$ [2], where $L_{drawn(\mu m)}$ is the minimal length of transistor gate in micrometers. The FO4 metric is a useful measure to estimate the processor clock speed across technology generations [3]. In the literature the access delay of some structures in the critical path are used as a lower bound of the microprocessors clock cycle time. [2] suggests using the computation delay of very optimized 64-bit adder (5.5FO4 for static logic), under the assumption that to execute two dependent instruction in consecutive cycles clock, the first instruction must compute its result in a single cycle. Considering the pipeline-latches overhead and time to bypass the adder result back to the input for the next instruction, clock period is goes down to 8 FO4 delays in aggressive estimates and 16 FO4 delays in conservative estimates. With these considerations, the FO4 delay decreases from 64ps in an 180nm technology to, while frequency clock increases from 1.09GHz in an 180nm to 5GHz in a 70nm Technology.

3.1 Wire considerations

The source of technology parameters used in this work is Predictive Technology Model (PMT) provided by the device group at UC Berkeley [1]. The model uses the same approach as the International Technology Roadmap for Semiconductors by subdividing the wiring layers into three categories 1) Local lines, for connections within a cell, 2) Intermediate lines for connections between modules and 3) Global lines for chip communications. In addition, it use two structure types a) Global Layer lines are coupled above one metal ground and b) Local and Intermediate Layers lines are coupled between two metal ground.



a) Global Layer lines b) Local and Intermediate Layers lines

Fig. 2. Wire Structures used in PMT

Table1 shows the wire parameters for different technology generations. R*wire*, C*couple*, C*ground* and C*total* values are based on the analytical model of PMT [1]. Here we can see that while the transistor's speeds are scaling more or less linearly, wires are getting slower

with each new technology generation. This is because the resistance of wires increases with each new technology generation but the capacitances stay more or less constant.

Table 1. Typical Wire Dimension and its parasitic effects

	Widt (µm)	Space (µm)	Thickness (µm)	Heightid (µm)	Kild (µm)	Rwire Ω/µm	Cc fF/µm	Cg fF/µm	Ct fF/µm
0.180 µm Technology generation @ 1.9 GHz									
Local	0.28	0.28	0.45	0.65	3.5	0.174	0.075	0.030	0.212
Intermediate	0.35	0.35	0.65	0.65	3.5	0.096	0.078	0.038	0.233
Global	0.80	0.80	1.25	0.65	3.5	0.021	0.081	0.080	0.243
0.130 µm Technology generation @ 2.7 GHz									
Local	0.20	0.20	0.45	0.45	3.2	0.244	0.089	0.029	0.236
Intermediate	0.28	0.28	0.45	0.45	3.2	0.174	0.060	0.040	0.201
Global	0.60	0.60	1.20	0.45	3.2	0.030	0.089	0.079	0.258
0.100 µm Technology generation @ 3.47GHz									
Local	0.15	0.15	0.30	0.30	2.8	0.488	0.068	0.028	0.193
Intermediate	0.20	0.20	0.45	0.30	2.8	0.244	0.068	0.037	0.212
Global	0.50	0.50	1.20	0.30	2.8	0.036	0.088	0.082	0.256
0.070 µm Technology generation @ 5.0 GHz									
Local	0.10	0.10	0.20	0.20	2.2	1	0.053	0.022	0.152
Intermediate	0.14	0.14	0.35	0.20	2.2	0.448	0.057	0.031	0.177
Global	0.45	0.45	1.20	0.20	2.2	0.040	0.073	0.082	0.230

3.2 Wire delay

Since the delay time is proportional to the product of resistance and capacitance, evaluated wire delays for various clock rates using Spice3 for different technologies: 180nm @1GHz, 130nm @2GHz, 100nm @3GHz and 70nm @5Ghz. A distributed RC model with 1µm wire RC basic segment was used. Figure 3 shows the delay for local and intermediate wires expressed in FO4 delays, while Figure 4 shows Global wires delay and distributed RC model.

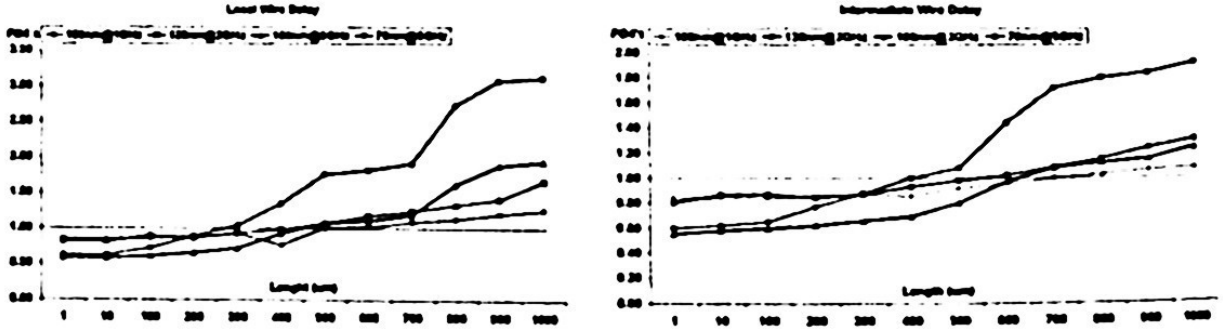


Fig. 3. Wire Delay evaluation using Spice3

For the load at the far end of each segment evaluated, we used a buffer with high spee characteristics: under the assumption that an optimal segment length makes the wire delay equal to or less than one FO4 delay. Wires with delay under one FO4 line in Figures 3 4 are considered optimal segment length without repeaters.

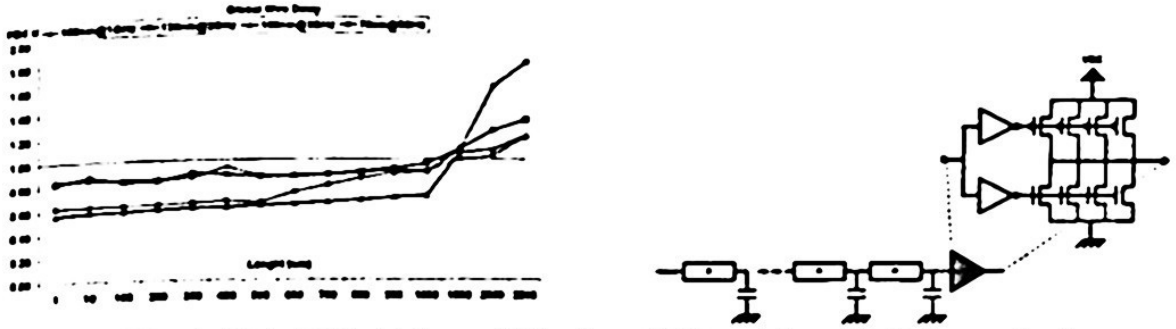


Fig. 4. Global Wire Delay and Distributed RC model used in Delay evaluation

3.3 Reducing Wire Delays

In long wires the insertion of repeaters periodically along the wire brakes the quadratic function of the delay on the wire length and makes it a linear function of the total wire length. The general expression for delay in a wire of length L with distributive RC is given by $T_p = 0.38 \times R \times C \times L^2$, while in a wire with M-1 repeaters the expression for delay is $T_p = 0.38 \times R \times C \times \lceil L/M \rceil \times M + (M-1)_{t_{buffer}}$ where t_{buffer} is the buffer delay time and M the number of wire segments. The optimal number of buffers on total wire length is obtained taking the first derivative with respect to segments number M, $\partial Tp/\partial M = 0$ then the number of repeaters is $M = L(0.38 \times R \times C / t_{buffer})^{1/2}$, taking into consideration $M=1$ for a wire with length $2l$, the maximum wire length tolerable without have need of repeaters theoretically is $l = 0.5\sqrt{(t_{buffer} / 0.38RC)}$.

With these considerations we designed the CAM and RAM memories for 70nm technology, as explained in the following sections.

4 Microarchitecture of Circuits

In order to evaluate timing and power of the issue queue, we considered 70nm technology parameters. The size of a memory cell is basically determined by the wires. For bit-line and word-line size we used intermediate wire parameters, interconnections between transistors are made with local wires. The length of wires in our designs is below the optimal segment length. Note that we evaluated only the access time of structures and did not consider the latencies of allocation logic and selection logic.

4.1 CAM comparators

The model for a CAM is shown in Figure 5 and its operation is as follows. Each time an instruction is written in the queue, the word line's OR resets the ready bit flip-flop and turns on the transfer gates to enable the precharge of Match Lines. In the comparison stage this precharge signal is driven high in the same clock that Tag bits are driven on each comparison port. If the data stored in the cell is equal to the tag driven on the comparison port(s), the mach line keeps the high level, otherwise it is discharged. The match line's OR sets the OpRdy bit flip-flop and the negative output disables the Match Line corresponding to this entry to save the precharge energy for posterior comparisons. If operands are available in allocation stage FF for ready logic is set on allocation stage.

CAM-wakeup model uses duplicated CAM with *EW* comparison ports for each source operand, while Dependency-Matrix-wakeup model use a single CAM with *2DW* comparison ports, where *EW* and *DW* are Execution Width and Decode Width, respectively. We evaluate circuits for both models of CAM wakeup with six comparison ports and Matrix wakeup with eight comparison ports.

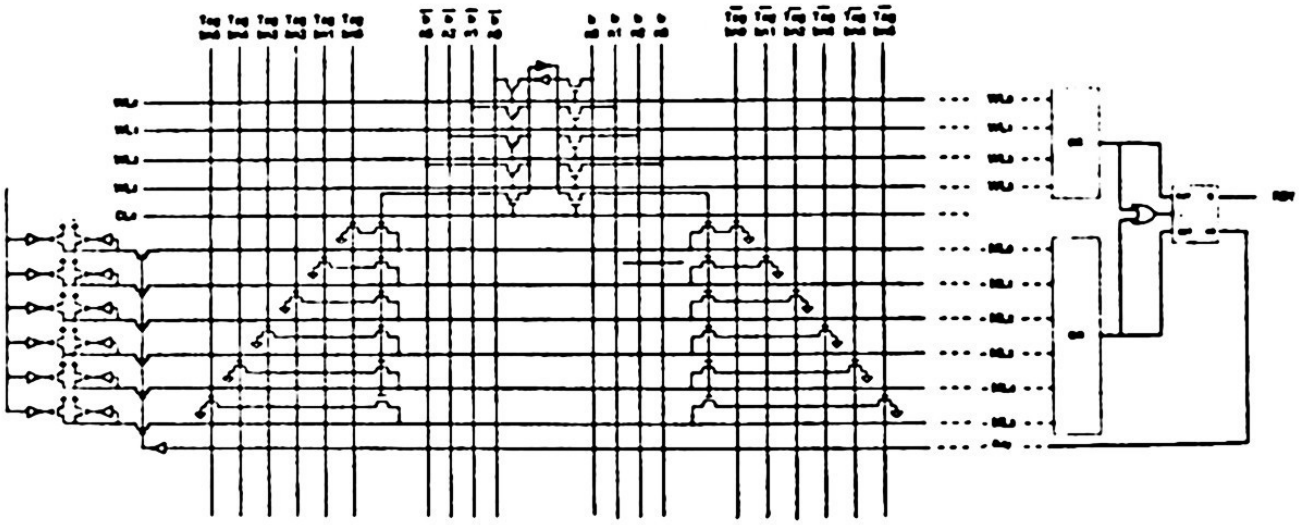


Fig. 5. A CAM Entry with 4 Write ports and 6 Comparison Ports
Basic Cell Area *(lxh)*:4W4C=4.48μm *X* 2.80μm, 4W6C=5.6 μm *X* 3.36μm

The CAM size is 32 by 7-bits. It also contains the ready register and 64X7-bits plus ready register for 32 entries and 64 entries instruction queues, considering a processor with register files of 128 entries. Figure 6 displays the write cycle, comparison cycle, flip-flop timing and average power by access, referenced by the clock signal (dashed line the graphics). Delay for Ready logic is around 100ps, delay for comparison time is taken since the precharge is driven to signal MTCH is generated, delay for write access is taken since the data is driven (driver data delay is added) to data is modified and stabilized the ram-cell. Power consumption measure technique is explained in section 6.

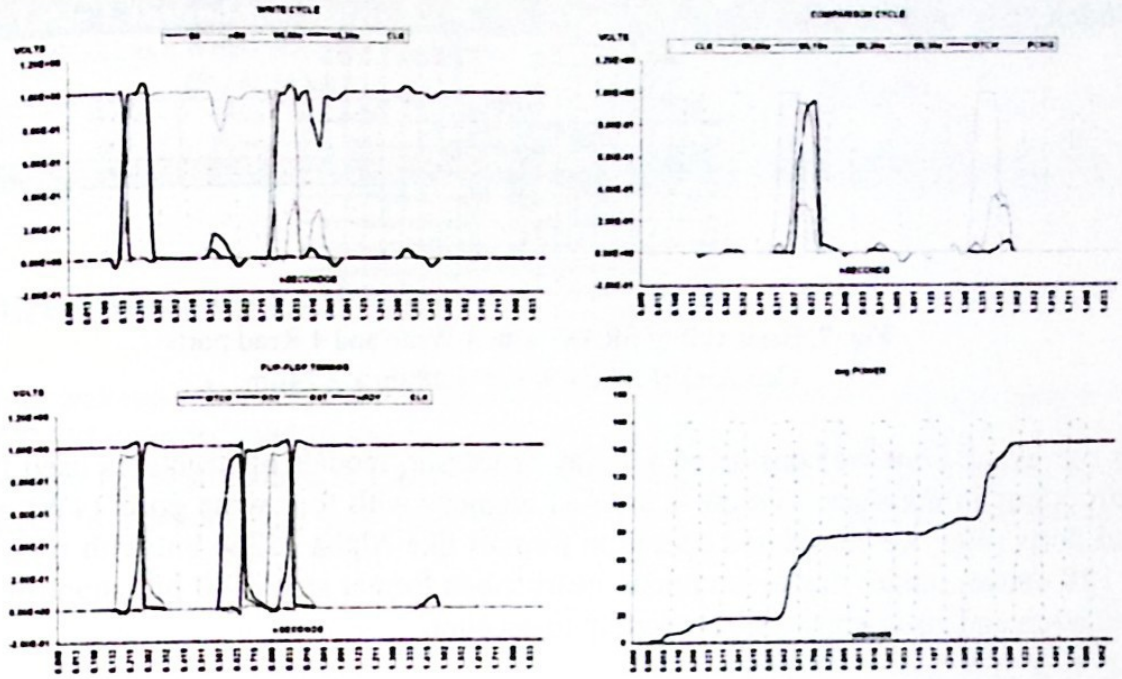


Fig. 6. Wave form of write cycle, comparison cycle, flip-flop timing and average power of CAM with 32 entries 4 Write ports and 6 Comparison ports

Table 2 shows the measure of timing and power from spice3, for timing measure the biggest delay between bit-line and word-line was taking (considering drivers delay). For write access, basic-cell stabilization delay was added and for comparison, MTCH signal generation delay was added. Power was measured on complete structures.

Table 2. CAM 70nm technology

CAM entries	7 bits 4 Write Ports, 6 Comparison Ports(l=31.36μm)					7 bits 4 Write Ports, 8 Comparison Ports (l=39.20μm)				
	$h(\mu m)$	Write (ps)	Avg Power	Comp (ps)	Avg Power	$h(\mu m)$	Write (ps)	Avg Power	Comp (ps)	Avg Power
32	89.6	42.8	18.80mW	73.9	47.15mW	107.52	48.6	76.95mW	104.9	94.45mW
64	179.2	55.5	36.50mW	80.0	95.27mW	215.04	56.3	153.55mW	130.7	166.95mW

4.2 SRAM Issue Window

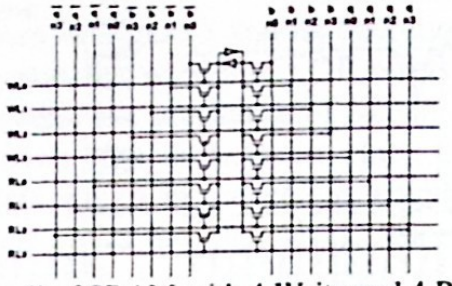


Fig. 7. Basic cell of SRAM with 4 Write and 4 Read ports
Cell Area ($l \times h$): 4W4C= 4.48µm ×2.24µm

For our evaluations we considered a 4-way processor, models of structures used for store instructions in the issue window is a RAM memory with four write ports (4W) and four read ports (4R), we considered operation formats like Alpha 21264 but with register files of 128 entries, under this assumptions instructions format size is 40 bits, control bits for allocation and issue logic are not taken in to account.

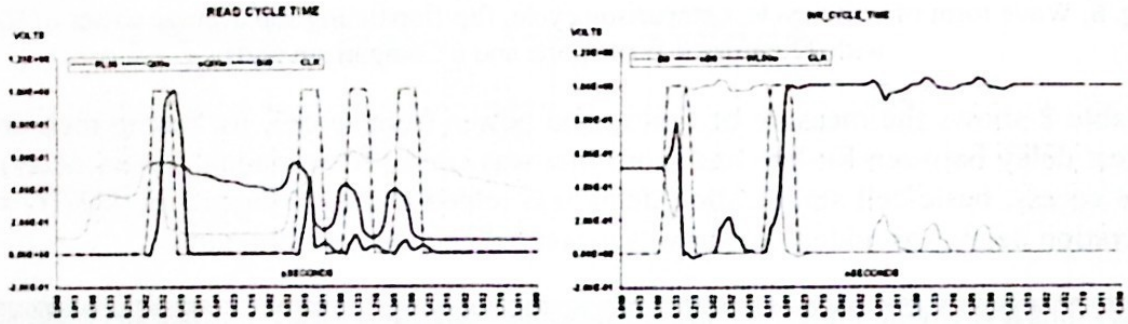


Fig. 8. Waveform of read cycle and write cycle time of RAM with 32 entries
4 Write ports and 4 Read ports

Table 3. Issue Queue RAM 70nm technology					
RAM entries	40 bits 4 Write Ports and 4 Read Ports *(l=156.8 µm)*				
	h(µm)	Write *(ps)*	Avg Power	Read *(ps)*	Avg Power
32	71.68	56.5	6.14 *mW*	22.5	11.50 *mW*
64	143.36	71.5	9.85 *mW*	22.5	13.75 *mW*

4.3 RAM Dependency Matrix.

Dependency matrix structure is a RAM like shown in figure 7; this is a square matrix of $N \times N$ where N is queue size.

Table 4. Dependency Matrix RAM 70nm technology									
32X32 bits 4 Write Ports and 4 Read Ports *(l=143.36 μm)*					64X64 bits 4 Write Ports and 4 Read Ports *(l=286.72 μm)*				
h(μm)	Write *(ps)*	Avg Power	Read *(ps)*	Avg Power	*h(μm)*	Write *(ps)*	Avg Power	Read *(ps)*	Avg Power
71.68	48.6	6.94mW	22.5	8.90mW	146.36	65.5	19.00mW	22.5	22.00mW

5 Direct Wakeup Evaluation

Direct wakeup schema is show in figure 10. The goal of this mechanism is replace the use of CAM memory used in associative searches on wakeup stage by the use of pointer keep in two RAM structures, a Mapping table **MT** and a little Dependency Matrix **MWT** to wakeup instructions with multiple-dependents . Direct Wakeup Mechanism is proposed in [7].

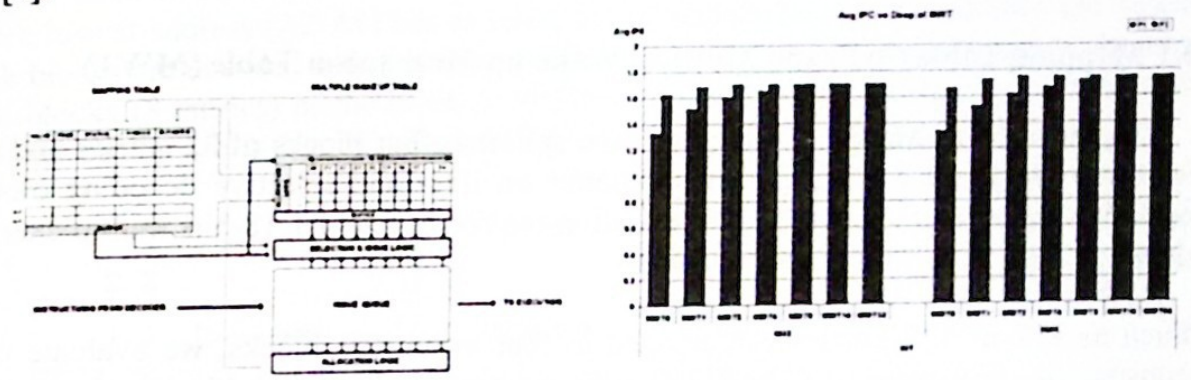


Fig. 9. Scheme of direct Wakeup Mechanism and Impact of MWT deep on performance

This MT allocates in each entry follows fields (example for one dependent-instruction-pointer): The free register bit, the value register (both two last field are part of register file), the status register, the consumer instruction pointer and a Pointer to Multiple-Wake-up-mechanism table. The status stage could be 00: No-dependent, 01: One dependent 10: Multiple-dependents, 11: Computed Value (Ready), for this case. The pointer for multiple wakes up mechanism is used depending of the third field value –If the register is not free and the status is 10: Multiple dependents. Mapping table is an extension of register file but with independent ports for reads, used on wakeup stage. Allocation of pointer is performance on decode stage, at checking time for ready operands. Wakeup is achieved reading the MT indexed by destination register file and reading MWT indexed by M-POINTER if instructions with multiple-dependents exist.

In order to evaluate the microarchitecture behavior of Direct-Wakeup scheme, we used SimpleScalar 3.0. the simulator has been modified to model separately integer, floating

point and load/store queues, and was build for Alpha configuration, as a workload. SPEC2000 programs was simulated (each of which ran for 200 million instruction). Figure 9 shows the impact of MWT deep on performance processor; right columns each group correspond to the maximum IPC possible for its configuration. MWT with entries using single pointer, experiment IPC loss of 4.8% for FP and 2% for INT. MWT with 4 entries, experiment IPC loss of 10% for FP and 4% for INT benchmark respectively.

For Spice3 simulation we use MWT with 16, 8, 4, 2 entries and single pointer. Considering 4-way Out Of Order processors, these structures require 8 write ports and read ports for MT and MWT. The status fields are independent state-machines with parallel inputs which can do change the state of machine (inputs Ored) on each access each physical register. The MWT, needed 8 1-bit write ports, and 4 full-bit (32/64) read ports.

5.1 Mapping Table (MT) and Multiple Wake-up Mechanism Table (MWT)

Organization of Mapping Table has been split into four blocks of 32 entries and these in 4 sub blocks of 8 entries, causing shorter bit lines and selective decoding block optimize power and cycle time. Organization require of one 4:1 18-bits multiplexer with delay of 22ps, for a single output data bus.

Such as MT of 128 Entries was divided in four symmetric blocks, we evaluate power consumption for writes (2.128*mW*/2.59*mW*) and reads (3.058*mW*/3.19*mW*) of a single entries per 10/11-bits block respectively, measures of only one cycle without activity report around of 0.3mW. In order to evaluate total power we have considered access power plus three times power on idle stage.

Table 5. MT RAM 70nm technology

RAM 128 4 Banks of 32 Entries	10bits 8Write Ports and 6 Read Ports (l=78.4 μm)				
	$h(\mu m)$	Write(*ps*)	Avg Power (*mW*))	Read (*ps*)	Avg Power (*mW*))
	125.44	56.5	3.028 *mW*	45.0	3.958*mW*
	11bits 8Write Ports and 6 Read Ports (l=86.24 μm)				
	125.44	56.5	3.49*mW*	45.0	4.09*mW*

Multiple Wake up Table MWT is a RAM structure of 2, 4 or 8 entries, 8 write ports decoding the IQ pointers to write one bit in the corresponding entry which is reserved allocation logic, write access is validated by the status signals latched in the register read stage. 6 read ports are necessary in case of maximum possible number of instructions execution have it all multiple dependents.

Table 7. MWT RAM timing and power for 70nm technology

MWT entries	32bits, 8 Write and 6 Read Ports (l=250.88 μm)					64 bits, 8 Write and 6 Read Ports (l=501.76 μm)			
	h(μm)	Write (ps)	Avg Power	Read (ps)	Avg Power	Write (ps)	Avg Power	Read (ps)	Avg Power
1	3.92	23.6	1.00 mW	22.0	4.70 mW	23.6	5.00 mW	22.0	9.51 mW
2	7.84	23.6	1.30 mW	22.0	4.73 mW	23.6	8.16 mW	22.0	9.50 mW
4	15.68	23.6	4.97 mW	22.0	7.10 mW	23.6	8.83 mW	22.0	13.42mW
8	31.36	25.6	7.25 mW	22.0	8.93 mW	25.6	10.53 mW	22.0	15.05mW
16	62.72	38.2	10.50 mW	25.0	11.50mW	38.2	15.30 mW	25.0	19.43mW

5.2 Decoder

Such as MT-memory is splitting into 4 blocks, decoder is designed using dynamic gates to optimize timing and power the address word is split to reduce the number of decode stages. Each basic decoder uses 8 row decoder like show in the figure 12c, decoding the three lowest address (A2-A0) bits to select one of eight entries of the memory sub block. Sub blocks are decoded using two next address bits (A4-A3) to selecting one of the four sub blocks (8 entries) of the block array (32 entries). Finally blocks are decoded with another similar 2 bits decode using (A6-A5).

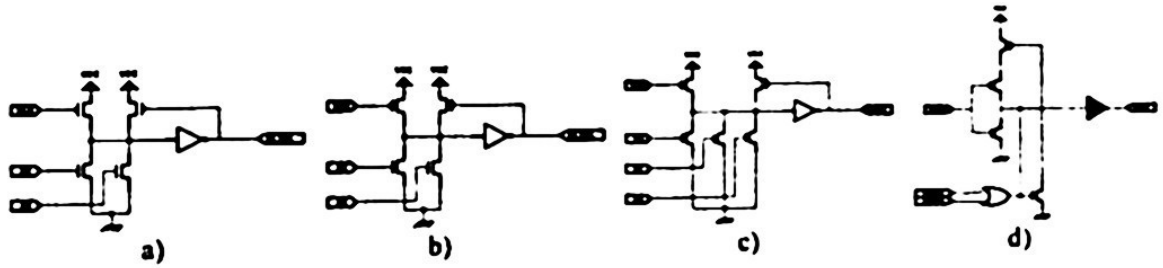


Fig. 10. a) Block decoder b) Sub block decoder c) ROW decoder and d) Enable WL circuit

The enable word line circuit shown in figure 12d is used to isolate the RAM array of the decoder. Selection block and Selection sub block signal enabling or disabling the WL. Table 6 show the delay and consumption of 7-bits address decoder.

Table 6. 7 Address bits DECODER 70nm technology

DECODE	7 Address bits Decoder		
	h(μm)	Delay(ps)	Avg Power (mW)
128WL	125.44	32	8.5

6 Power Evaluation

Power dissipation is one of the most important factors on evaluation of VLSI designs [8], new approaches in low power require of accurate power consumption simulation new technology generations. Note that in general expression for power, the consumption depends strongly of the circuits capacitance $p = C_T \cdot V_{DD}^2 f$, however estimation of C_T requires not only identification of state-changing in logic gates, but also the effective capacitances in gate regions and drain regions a long with biasing effects, this is the analogical behavior of transistors. In order to achieve an accurate monitoring of power dissipation in VLSI circuits, we use Spice3 simulator tool. There is a sub-circuit to measure average power taking advantage of the voltage equation on one capacitor (see figure 11).

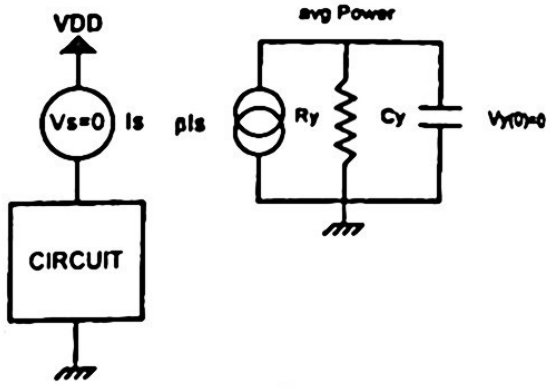


Fig. 11. Power meter

This meter is an independent subcircuit with a current-controlled current source and a parallel RC circuit. Power average in a dissipative element with fixed source voltage V_{DD} in a time interval $\Delta t = t_2 - t_1$ is given by:

(1) $P_{avg} = V_{DD}/t_2 - t_1 \int_1^2 i_{DD}(t)dt$ by using a current-controlled current source with current equal to $i_x = i_{DD}$, Voltage in a Capacitor C_y (from figure 11) in a time interval $\Delta t = t_2 - t_1$ is given by next equation

(2) $V_{C_y}(t) = \beta/C_y \int_1^2 i_{DD}(t)dt + v_o$ by running a transient analysis and reading V_{C_y} at time t1 and t2 the average power consumption of the circuit is monitored by the equation 2 if a value for β is chosen such that

(3) $\beta = C_y V_{DD}/t_2 - t_1$ or $\beta = C_y V_{DD} f$.

The time interval $t_2 - t_1$ should span a clock period or multiple integer of period of the circuit frequency operation. Measuring the power dissipation is equivalent to measuring the supply current flow during the transient analysis. The current pulses are very short, and the current waveform must be computed carefully to accurately compute the power. In order to produce better results we choice second order gear rather than trapezoidal integration method.

7 Summary and Conclusions

Power and timing evaluation is an important factor on evaluation in new proposals, in this work we evaluated traditional Issue Queue instruction wakeup mechanisms and compared its results with a direct wakeup scheme proposed in [7], Table 8 shows results of these evaluations. In order to evaluate the total impact of structures in timing [IQ32/IQ64] and power (IQ32/64IQ) - for instruction issue queue for 32- and 64-entries respectively - On overlapped access to different structures maximum time delay is considered, but power is supplementary.
The average access power was computed taking into account all tree components of IQ operation: Allocation, issue, and Wakeup for all three mechanisms presented, values in brackets represent access time and values in parentheses represent access power to mentioned structures, for 32 and 64 entries respectively.
For CAM wakeup mechanism: on allocation stage writes of two source operands in CAM4W6C's [42.80*ps*/55.50*ps*] (18.80*mW*X2/55.50*mW*X2) and writes of the complete instruction in SRAM IQ are performed [56.50*ps*/71.50*ps*] (6.14*mW*/9.85*mW*). On Wakeup stage, comparison on two structures CAM are performed [73.9*ps*/80.0*ps*] (47.15*mW* X2/95.27*mW*) one for each operand in the IQ. On Issue stage instructions are read from SRAM IQ[22.5*ps*/22.5*ps*] (11.50*mW*/13.75*mW*).

For Dependency Matrix Wakeup mechanism: on allocation stage comparison of destination register are performed in one structure CAM4W8C[48.60*ps*/56.30*ps*] (76.95*mW*/153.55*mW*), writes of the complete instruction in SRAM IQ [56.50*ps*/71.50*ps*] (6.14*mW*/9.85 *mW*) and writes of dependency bits on matrix at time are performed [48.60*ps*/65.50*ps*] (6.94*mW*/19.00*mW*). On issue stage instructions are read from SRAM IQ [22.5*ps*/22.5*ps*] (11.50*mW*/13.75*mW*) and vector bit of matrix is read [22.5*ps*/22.5*ps*] (8.90*mW*/ 22.00*mW*) to wakeup instruction on next cycle.

For Direct Wakeup mechanism: we consider that in decode stage is known the address for allocation of instructions in the queue and processor check the register file for ready operands indexing it with the source operands of instructions. Then, on allocation stage writes of IQ-pointers in the MT [56.5*ps*/56.5*ps*] (3.028*mW*/3.49*mW*) after writes of M-pointers in the MWT 8-entries [25.6*ps*/25.6.6*ps*] (7.28*mW*/10.53*mW*) and writes of the complete instruction in SRAM IQ [56.50*ps*/71.50*ps*] (6.14*mW*/9.85*mW*) are performed, note that decode for write access is not necessary. On wakeup stage MT is indexed [32.00*ps*] (8.50*mW*) with the result-registers for read [45.0*ps*/45.0*ps*] (3.95/4.09*mW*) the IQ-pointer and MWT [22.00*ps*/22.00*ps*] (8.93*mW*/15.05*mW*) is indexed subsequent if instructions with multiple dependents are executed. On issue stage instructions are read from SRAM IQ [22.5*ps*/22.5*ps*] (11.50*mW*/13.75*mW*).

Table 8, shows the power and cycle time for all three models of instruction issue queue evaluated, cycle time and power is considered for each stage of issue logic, Direct Wakeup Scheme requires the longest cycle time for allocation and wakeup due for sequential access to the MWT, but this stages use half clock cycle. With Direct Wakeup Scheme save 67%/80% of total energy dissipated in a 32/64 CAM Wakeup Scheme.

Table 8. Issue Queue timing and power for 70nm technology

IQ entries	CAM Wakeup						
	Allocation (ps)	Avg Power$_A$ (mW)	Issue (ps)	Avg Power$_I$ (mW)	Wakeup (ps)	Avg Power$_W$ (mW)	Avg Power$_T$ (mW)
32	56.5	43.74	22.5	11.50	73.90	94.30	149.54
64	71.5	120.85	22.5	13.75	80.0	190.54	325.14
IQ entries	Dependency Matrix Wakeup						
	Allocation (ps)	Avg Power$_A$ (mW)	Issue (ps)	Avg Power$_I$ (mW)	Wakeup (ps)	Avg Power$_W$ (mW)	Avg Power$_T$ (mW)
32	56.5	90.03	22.5	11.50	22.5	8.90	110.70
64	71.5	182.40	22.5	13.75	22.5	22.00	218.15
IQ entries	Direct Wakeup						
	Allocation (ps)	Avg Power$_A$ (mW)	Issue (ps)	Avg Power$_I$ (mW)	Wakeup (ps)	Avg Power$_W$ (mW)	Avg Power$_T$ (mW)
32	82.1	16.44	22.5	11.50	99.0*	21.35*	49.32
64	97.1	23.87	22.5	13.75	99.0*	27.64*	65.26

*In this table we are considering the worst case, this is like all instructions having multiple dependents

References

[1] Ron Ho, Kenneth W. Mai, Mark A. Horowitz "The Future of Wires" proceeding of IEEE Vol.89, No.4, April 2001.

[2] Vikas Agarwal, Stefhen W. Keckler and Doug Burger "The effect of technology Scaling on Micro-architectural Structures" Tech Report TR2000-02

[3] S. R. Kunkel and J. E Smith "optimal pipelining in supercomputer" Proceeding of 13th International Symposium on Computer Architecture, 1986

[4] Masahiro Goshima Kengo Nishino, Yasuhiko Nakashima, Shin-ichiro Mori, Tashiaki Kitamura and Shinji Tomita "A High Speed Dynamic Instruction Scheduling Scheme for Superscalar Processors" proceeding of 34th ACM/IEEE International Symposium on Microarchitecture, December 1-5, 2001, pag. 225-236

[5] Daniel Folegnani and Antonio Gonzalez, "*Energy Effective Issue Logic*", Proceedings of 28th Annual of International Symposium on Computer Architecture, 2001. Page(s): 230-239, Göteborg Sweden.

[6] Marco A. Ramírez, Adrian Cristal, Alexander V. Veidenbaum, Luis Villa and Mateo Valero "A Simple Low-Energy Instruction Wakeup Mechanism" Proceeding of 5th International Symposium on High Performance Computing , Tokio-Odaiba, Japan. October 2003.

[7] Marco A. Ramírez, Adrian Cristal, Alexander V. Veidenbaum, Luis Villa and Mateo Valero "Direct Instruction Wakeup for OOO processors" Proceeding of International Workshop on Innovative Architecture for Future Generation High Performance Processor and Systems IWIA-2004.

[8] Sun Mo Kang, "*Accurate Simulation of PowerDisipation in VLSI Circuits*", IEEE Journal of Solid-State Circuits, vol. SC-21, No. 5, October 1996.

[9] Gregory J. Fisher, "*An Enhanced Power Meter for Spice2 Circuit Simulation*", IEEE Transaction on Computer-Aided Design. Vol.7 No. 5 May 1988.