A Low-Energy Memory Design Technique Based on Variable Analysis for Application-Specific Systems

Cao, Yun
Kyushu University

Yasuura, Hiroto
Kyushu University

出版情報：SLRC 論文データベース, 2002-04. Association for Computing Machinery
バージョン：
権利関係：

A Low-Energy Memory Design Technique Based on Variable Analysis for Application-Specific Systems

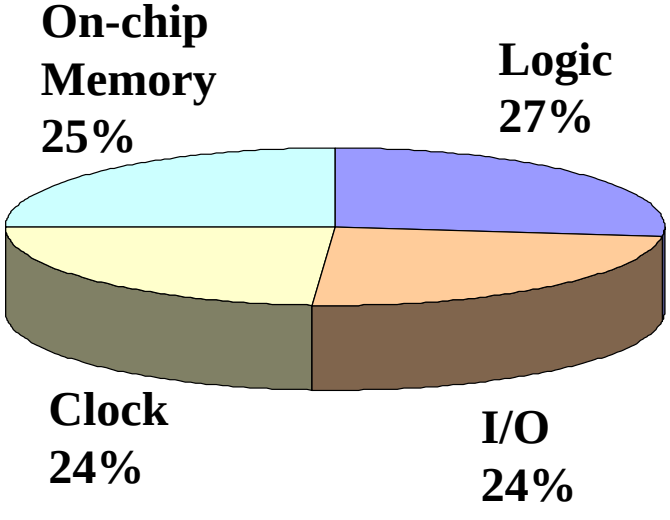
Yun Cao ***Hiroto Yasuura***

Kyushu University

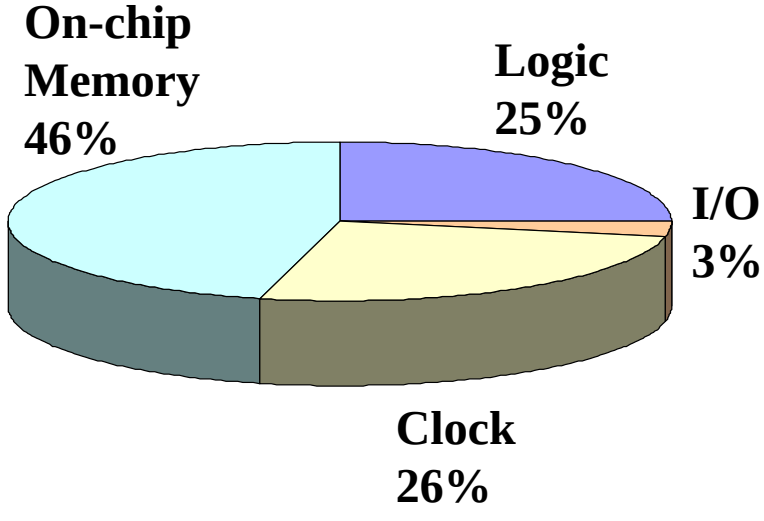
cao@c.csce.kyushu-u.ac.jp

Introduction

- The power consumption in memories can take a dominant fraction of the power budget of a whole embedded system for data-dominated applications (IMEC{Catthoor98} & {Catthoor01})

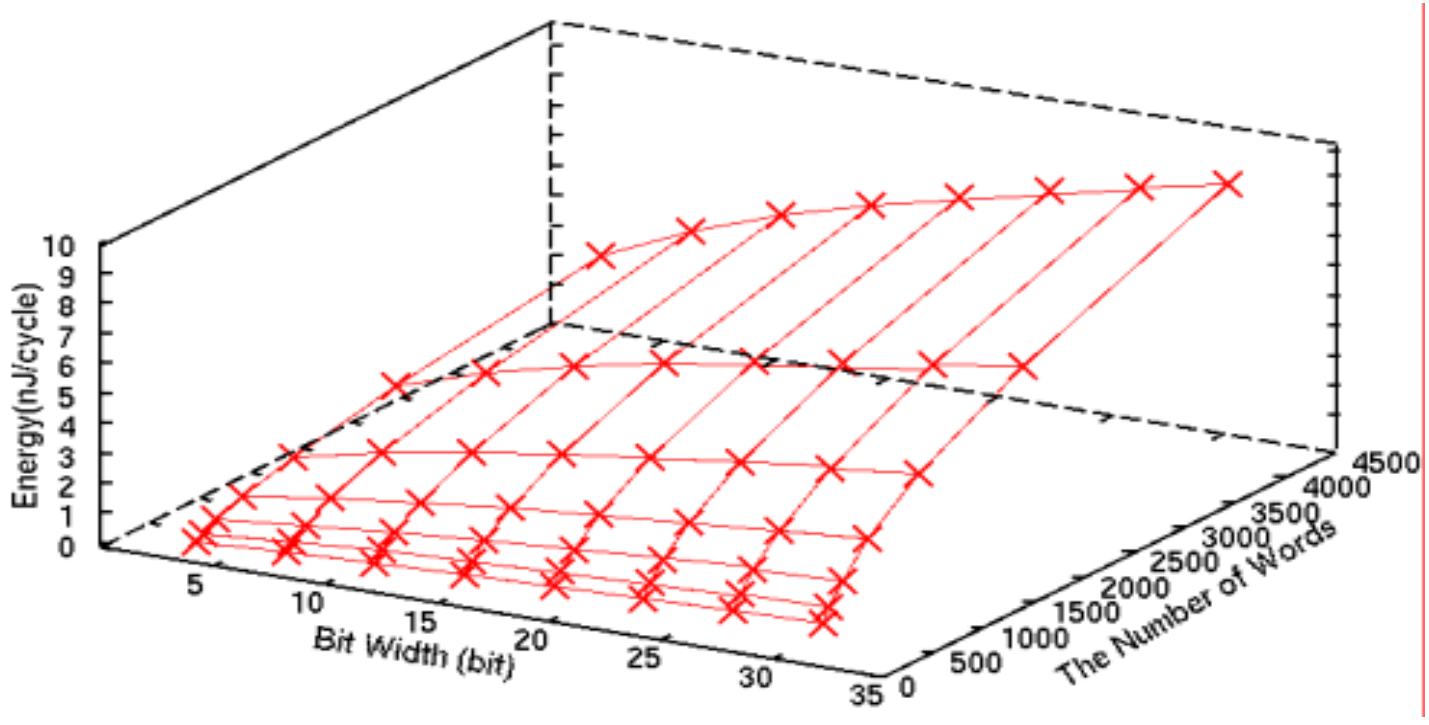


MPEG-2 Decoder



CPU with lots of caches

Size-energy correlation of memory



Bit width ***b*** and word count N_{words} v s. energy consumption of read access for SRAM

$$e_r = 24.9 \times \sqrt{b \times N_{words}} + 56[\,pJ/cycle\,] \quad (1) \qquad \text{Read access}$$

$$e_w = 197 \times \sqrt{b \times N_{words}} + 369[\text{pJ/cycle}] \quad (2) \qquad \text{Write access}$$

Variable analysis

- **Data width analysis**

  Effective data width *EWd* : the smallest data width which can hold both maximum and minimum values of a variable

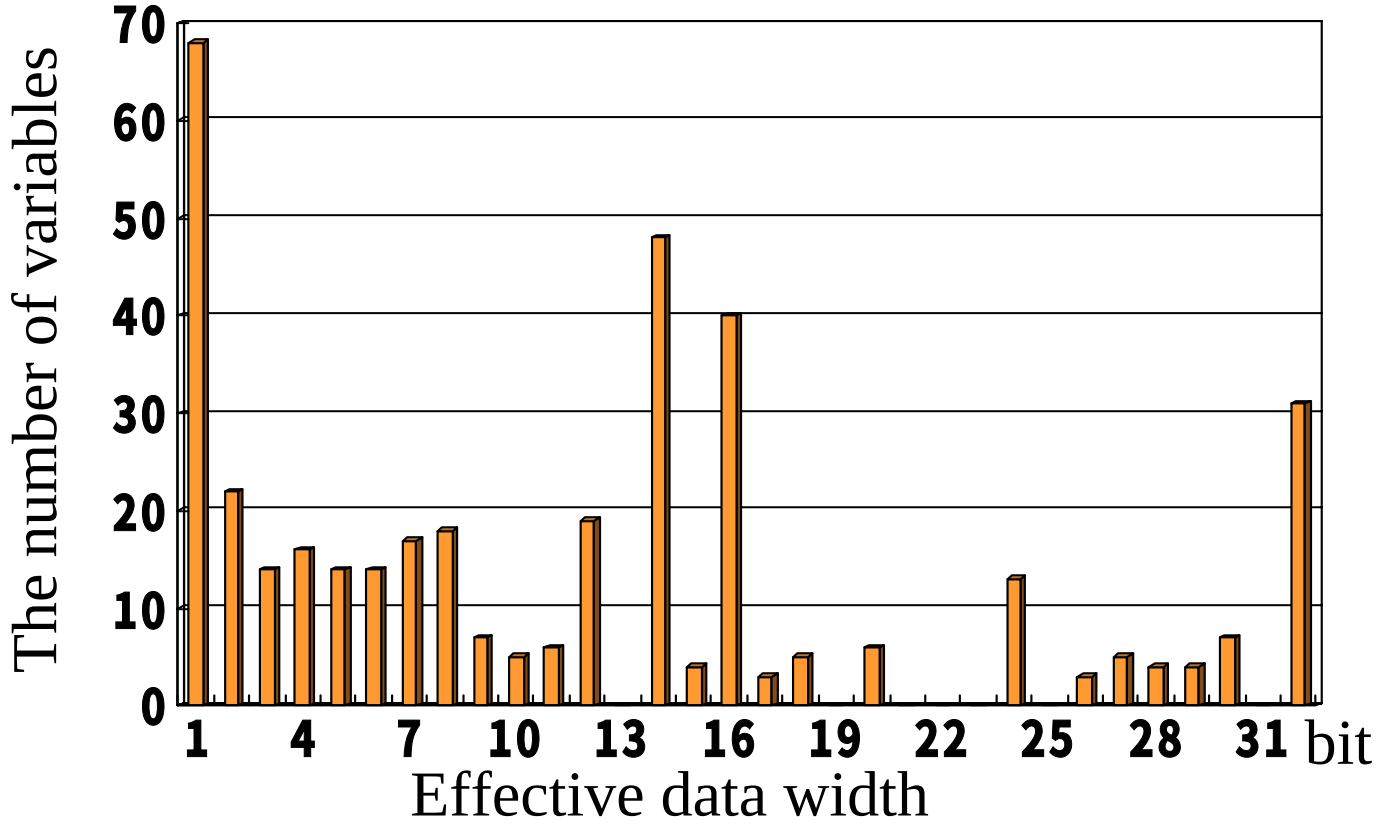
- **Life-time analysis**

  Life-time *LTx* : the period between the definition of a variable and its last use

- **Access frequency analysis**

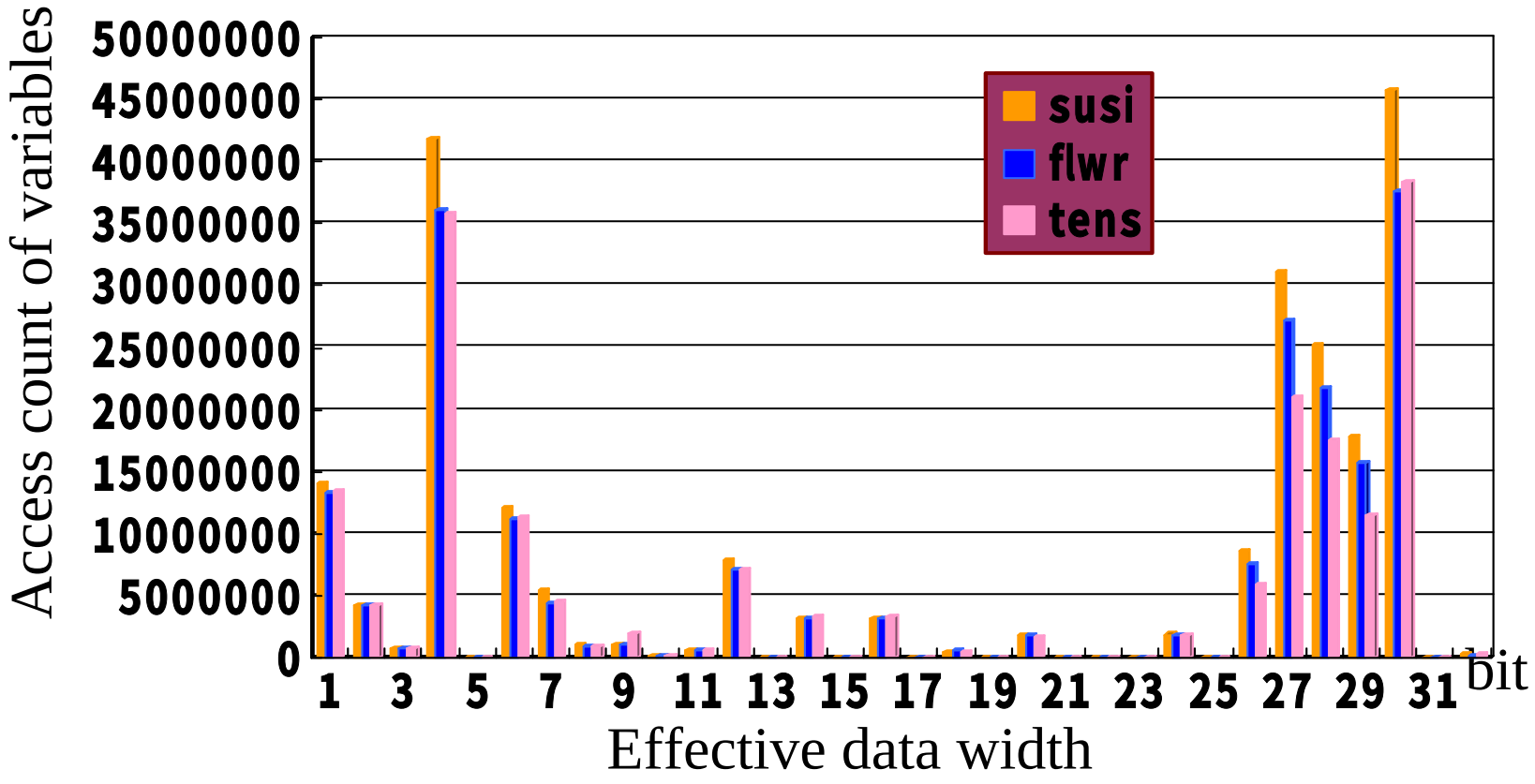
  Access count to a variable *TNa*

Variable analysis (Effective Size)



Results of data width analysis
for MPEG-2 video decoder

Variable analysis(Access frequency)



Results of access frequency analysis
for MPEG-2 video decoder

Low-Energy Memory Design

- Memory allocation:

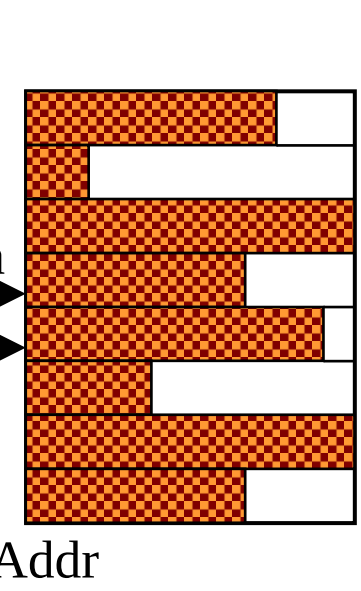
  Several memories are chosen from the available memory modules with different number of bit lines and word lines.

- Memory assignment:

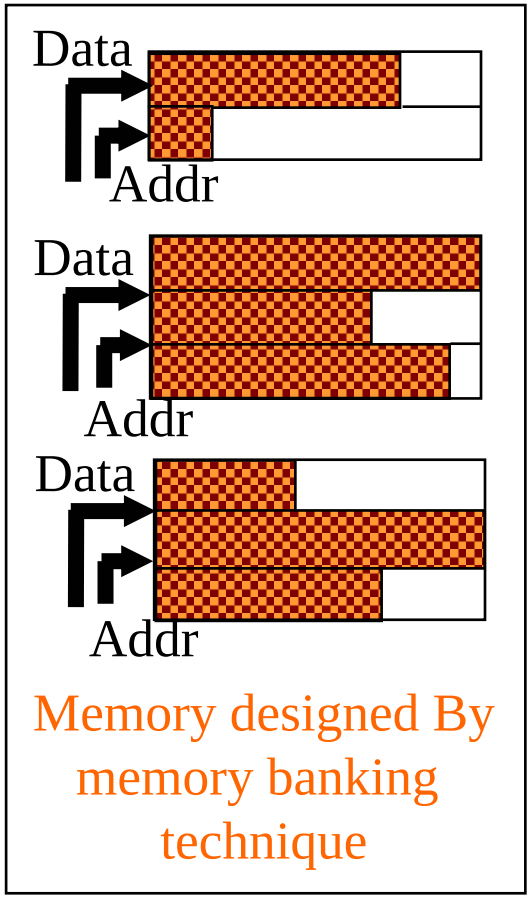
  The variables are assigned into those allocated memories.

In application-specific design, because the entire application is statically known, it let us perform memory allocation and assignment more intelligently to reduce the memory energy consumption

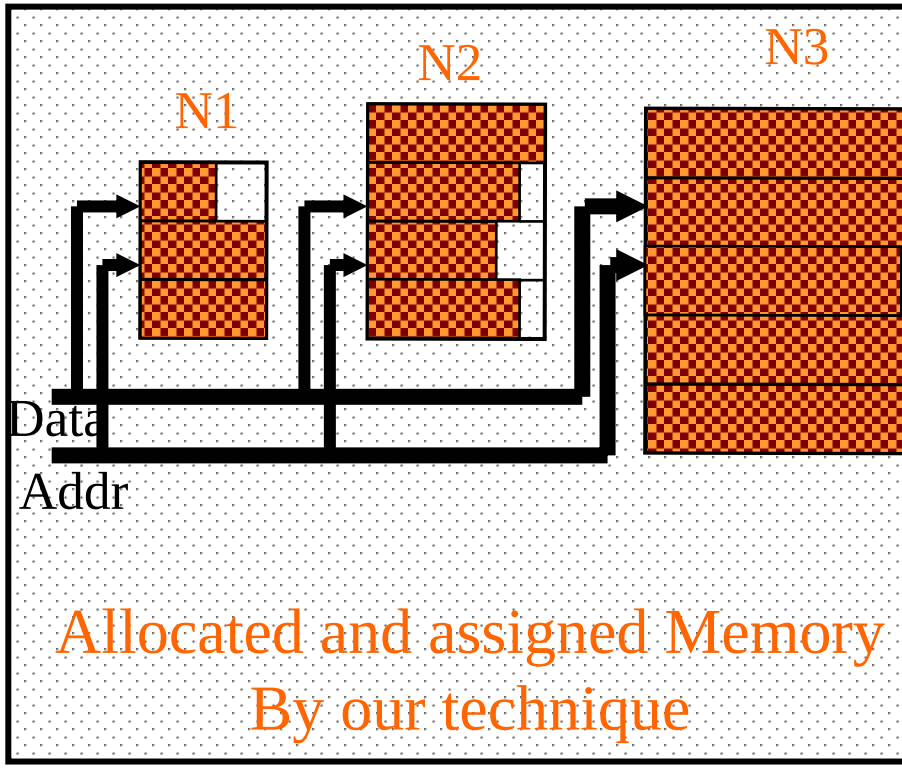
Basic idea



Monolithic Memory



Memory designed By memory banking technique



Allocated and assigned Memory By our technique

Variables with higher access frequency and smaller effective data width are assigned into a smaller low energy memory with fewer bit lines and word lines

Energy cost metrics

$$TEm = \sum_{N} (TEm(j) + TEm_{\delta}(j)) \qquad TEm(j) = \sum_{|Q(j)|} E(i, j)$$

$$E(i, j) = e_r(j) \times TNa_r(x_i) + e_w(j) \times TNa_w(x_i)$$

$$TEm_{\delta}(j) = Emon \times \delta(j)$$

TEm : Total energy consumption of memory

N : Total number of memory banks

$TEm(j)$: Energy consumption of memory bank j

$TEm_{\delta}(j)$: Energy overhead for added bank j

$Emon$: Energy consumption of a monolithic memory

$E(i, j)$: Energy consumption of x_i for read(write) access to bank j

X : A finite set of variables in a given application, $X = \{x_1, x_2, ..., x_i, ..., x_n\}$

x_i : A variable, $x_i \in X$

n : The number of variables in a given application program

$e_r(j)(e_{w(i)}(j))$: Energy consumption per read (write) access for bank j

$TNa_r(x_i)(TNa_w(x_i))$: The number of read (write) accesses for variable x_i

$Q(i)$: The set of variables assigned into bank j, $Q(j) \subseteq X$

$\delta(j)$: Overhead coefficient for added bank j , caused by addressing complexity

Problem formulation

Assumption

- We can generate arbitrary size of SRAM. The size of SRAM is limited by the total area available on-chip.
- Register allocation, which assigns frequently accessed variables such as loop indices to processor registers, has already been performed.

Given X, TNa, EWd, LTx, $N_{\max}$

To find N and $Q(j)$

So that $TEm = \sum_{N} (TEm(j) + TEm_{\delta}(j))$ is minimized

Subject to $N \le N_{\max}$, $Q(j) \subseteq X$

$X = Q(1) \cup Q(2),...,\cup Q(N)$, $Q(1) \cap Q(2),...,\cap Q(N) = \phi$

Allocated and assigned memory: N banks, $b(j) \times m(j)$ size

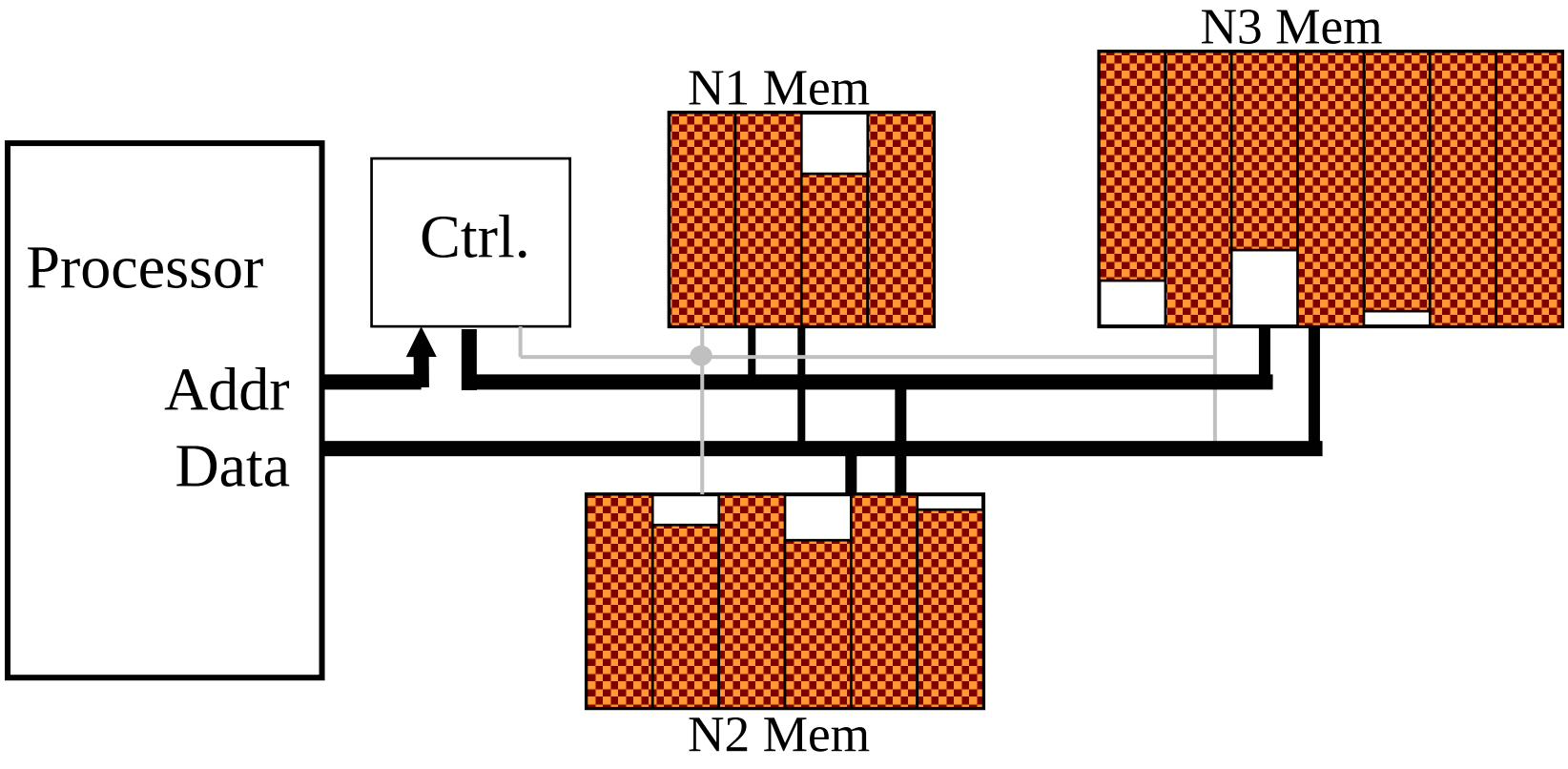
where $b = \max_{x_i \in Q(j)} EWd(x_i), m(j) = |Q(j)|$

Our approach

Our approach for low energy consists of the following phases:

- Analyze variables, report effective data width, life-time and access frequency.
- Perform initial memory allocation and assignment using analysis results including access frequency, life-time and effective data width of variables
- Change the number of banks, the variable assignment, under the constraints of the number of memory banks, and select the memory modules with different size including the bit width and the number of words to optimize the memory allocation and assignment in order to minimize memory energy consumption.

Experiments (Physical Model)



Assumption:

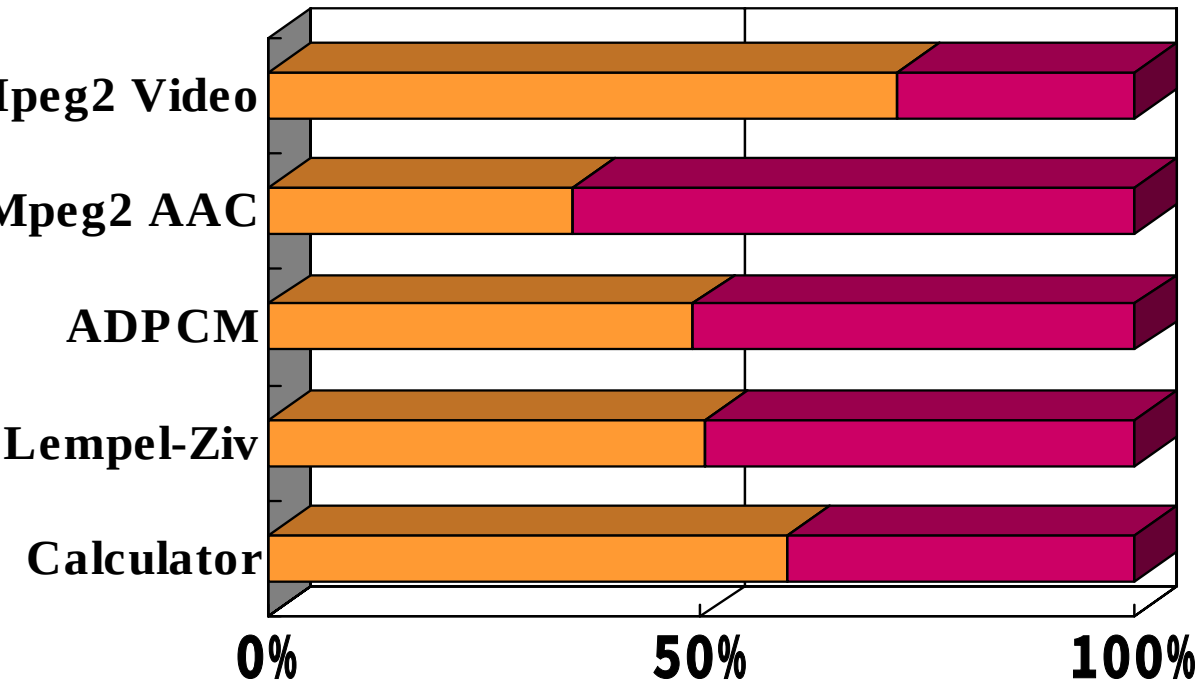
SRAM models with the arbitrary number of bit width, the arbitrary number of world lines, $N_{\max} = 3$, $\{\delta(1), \delta(2), \delta(3)\} = \{0.0, 0.15, 0.10\}$

Experimental results

Applications	Emon (J)	Memory banking Technique			Optimized Memory by VAbM		
		Configuration	TEb(J)	Sav.	Configuration	TEm(J)	Sav.
alculator	1.27 mJ	85 rows X 32b 154rows X 32b 533rows X 32b	0.87 mJ	31.5%	85rows X 8b 154rows X 32b 533rows X 32b	0.76 mJ	40.2%
empel-Ziv	1.37	830rows X 32b 3rows X 32b 1663rows X 32b	0.89	35.0%	830rows X 13b 3rows X 15b 1663rows X 15b	0.69	49.6%
DPCM	1.63	20rows X 32b 16rows X 32b 86rows X 32b	1.10	32.5%	20rows X 10b 16rows X 14b 86rows X 19b	0.80	50.9%
IPEG2AAC	1.05	30rows X 32b 2374rows X 32b 4804rows X 32b	0.39	37.1%	30rows X 20b 2374rows X 32b 4804rows X 32b	0.37	64.8%
IPEG2Video	145.1 kJ	26559rows X 32b 26557rows X 32b 28127rows X 32b	120.1 kJ	17.2%	26559rows X 8b 26557rows X 30b 28127rows X 32b	105.2 kJ	27.5%

Experimental results (1)

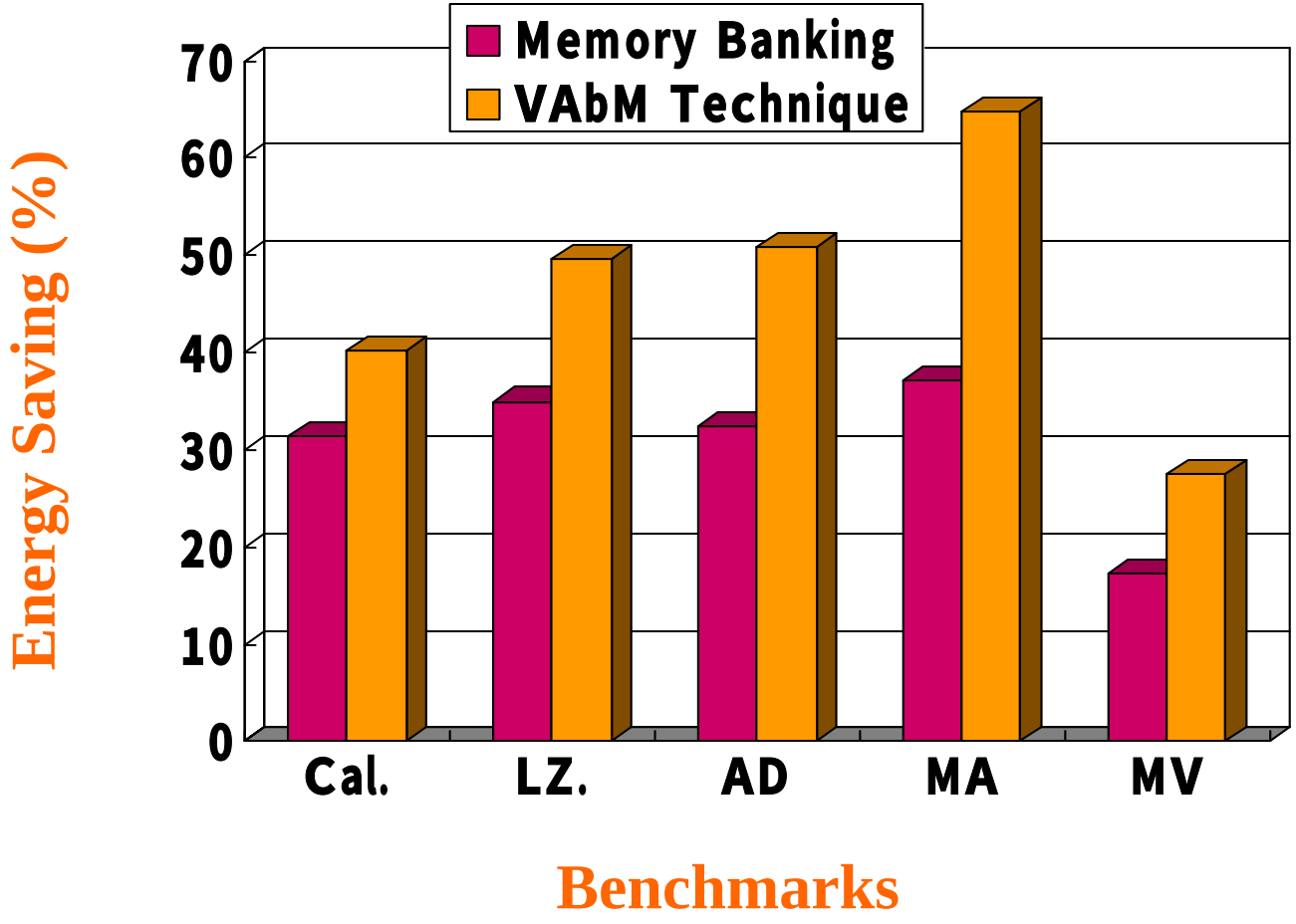
Benchmarks



Max 64.8% Reduction

Energy Consumption

Experimental results (2)



Conclusions

- Proposed a low-energy memory allocation and assignment technique based on variable analysis
- Demonstrated significant energy savings range from about 27.5% to 64.8%, based on several real embedded applications with respect to monolithic memory and 8.7% to 27.7% over memory designed by memory banking technique
- The hardware and wiring overhead due to additional memory banks is properly taken into account as a penalty factor