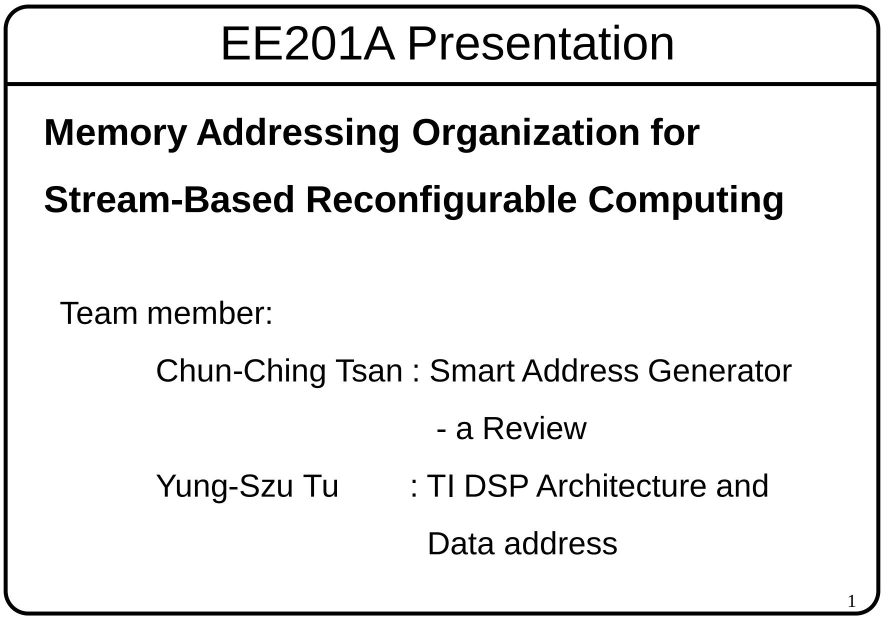
Reiner



EE201A Presentation

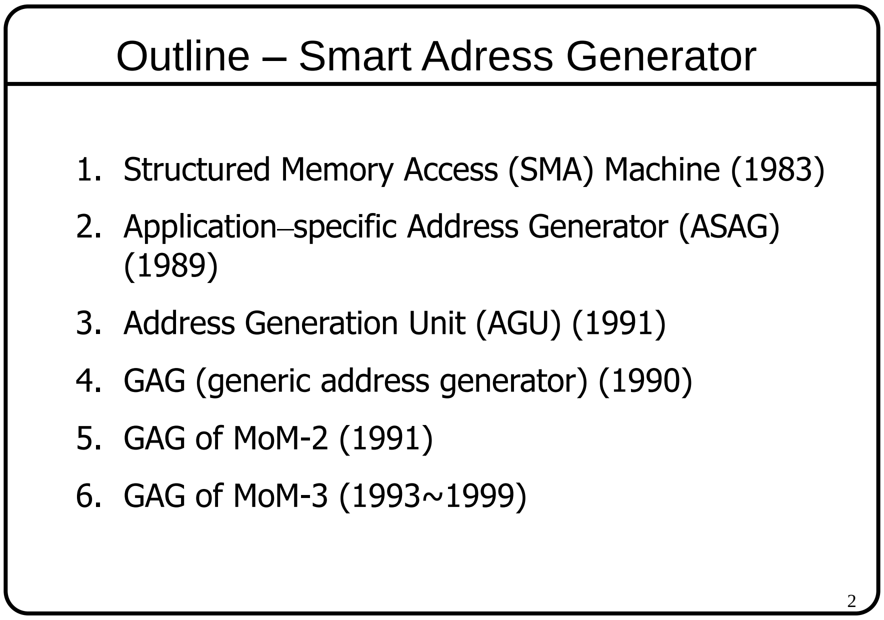
Memory Addressing Organization for

Stream-Based Reconfigurable Computing

Team member:

Chun-Ching Tsan	: Smart Address Generator - a Review
Yung-Szu Tu	: TI DSP Architecture and Data address

1



Outline – Smart Adress Generator

1. Structured Memory Access (SMA) Machine (1983)
2. Application–specific Address Generator (ASAG) (1989)
3. Address Generation Unit (AGU) (1991)
4. GAG (generic address generator) (1990)
5. GAG of MoM-2 (1991)
6. GAG of MoM-3 (1993~1999)

2

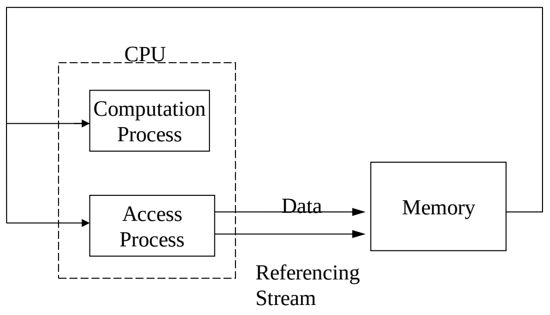
The GAG address generator:
How Ingrid' student eplains it

Reiner

Structured Memory Access (SMA) Machine (1983)

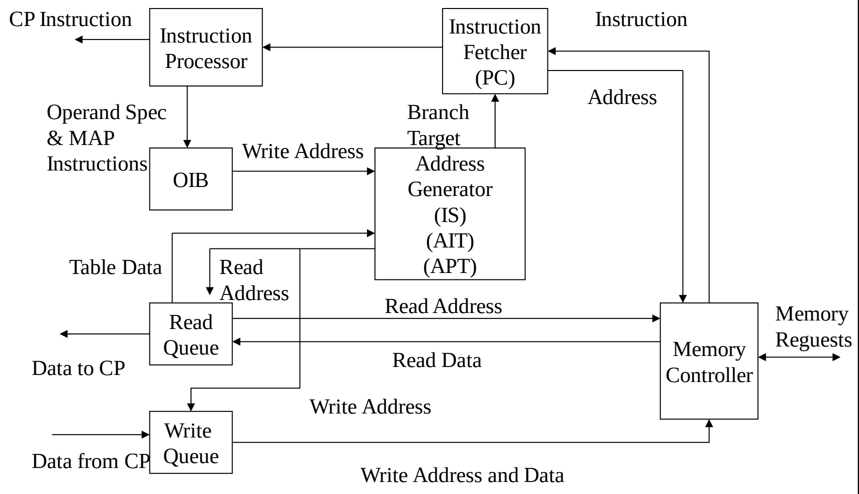
CPU-Memory Model: a von Neumann machine

- computational processor (CP)
- memory access processor (MAP)



3

MAP Internal Organization



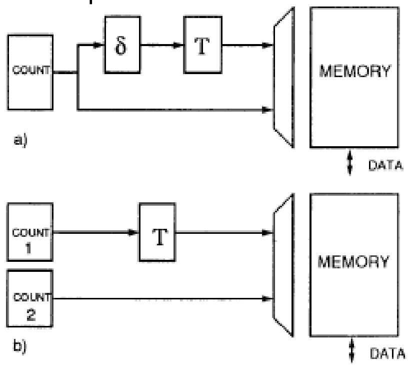
4

The GAG address generator:
How Ingrid' student eplains it

Reiner

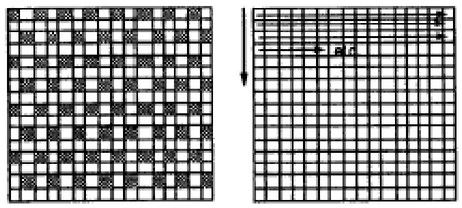
Application–specific Address Generator(ASAG) (1989)

- The needed address patterns are generated by a dedicated counters or circuit transformations applied to a counter output.



5

A Simple Example for Image Processing



```
for Y = 0 to 65535 step 4096,    (block height = 16 rows)
  for X = 0 to 255 step 16,      (block width = 16 columns)
    for i = 1 to 4,                                   (do 4 times)
      for y = 0 to 4095 step 512,                     (every 2nd line)
        for x = ( y / 512 ) mod 2 to 15 step 2,       (every 2nd pixel)
           address = x + y + X + Y,
        next x,
      next y,
    next i,
  next X,
next Y
```

6

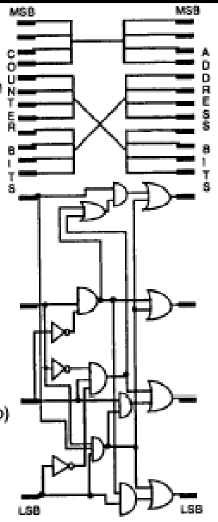
The GAG address generator:
How Ingrid' student eplains it

Reiner

Logic Synthesis for Semi-Random Address Sequences

```
(cbit0.cbit1bar.cbit2  +
cbit0.cbit2.cbit3)   ==> adbit 0
(cbit0bar.cbit1.cbit2bar  +
cbit0.cbit1.cbit2.cbit3)   ==> adbit 1
(cbit1bar.cbit2  +
cbit0.cbit2.cbit3)   ==> adbit 2
(cbit0bar.cbit1.cbit2bar.cbit3  +
cbit1bar.cbit2.cbit3  +
cbit0.cbit2.cbit3)   ==> adbit 3
cbit 8 ==> adbit 4
cbit 9 ==> adbit 5
cbit 10 => adbit 6
cbit 11 => adbit 7
cbit 4 ==> adbit 8
cbit 5 ==> adbit 9
cbit 6 ==> adbit 10
cbit 7 ==> adbit 11
cbit 12 => adbit 12
cbit 13 => adbit 13
cbit 14 => adbit 14
cbit 15 => adbit 15
```

(a)



(b)

7

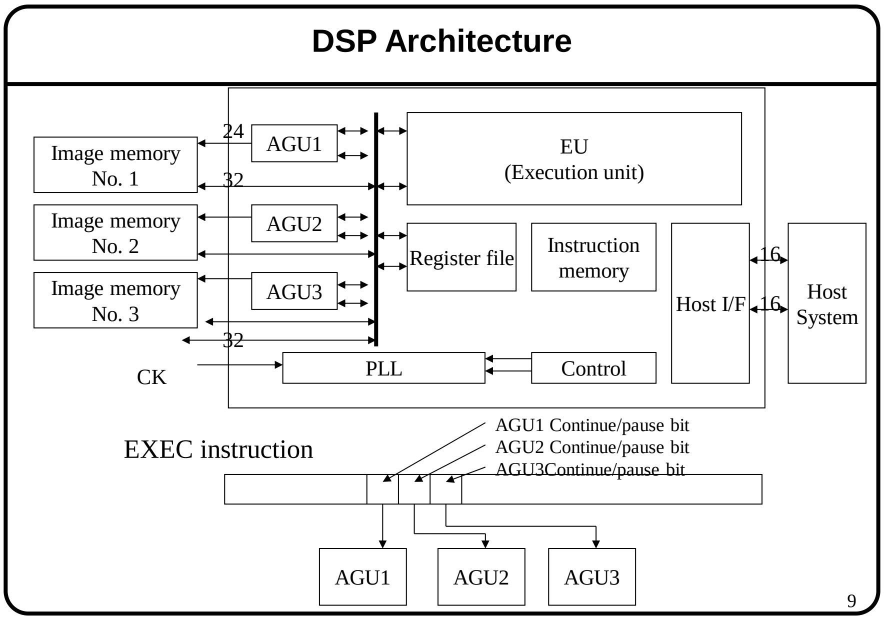
Address Generation Unit (AGU) (1991)

- an application specific address generation unit for video signal processor (VSP), a specified DSP
- implementing a 2-level address generation with window based memory access, without full slider method.
- 3 AGUs running in parallel calculate the address for external image memory
- Providing 17 addressing modes:
    - a 2-D raster scan mode
    - a block scan mode for spatial filtering
    - 8 variants of a neighborhood search mode
    - a 2-D indirect access mode for external image memory
    - a FFT mode and an affine transformation mode

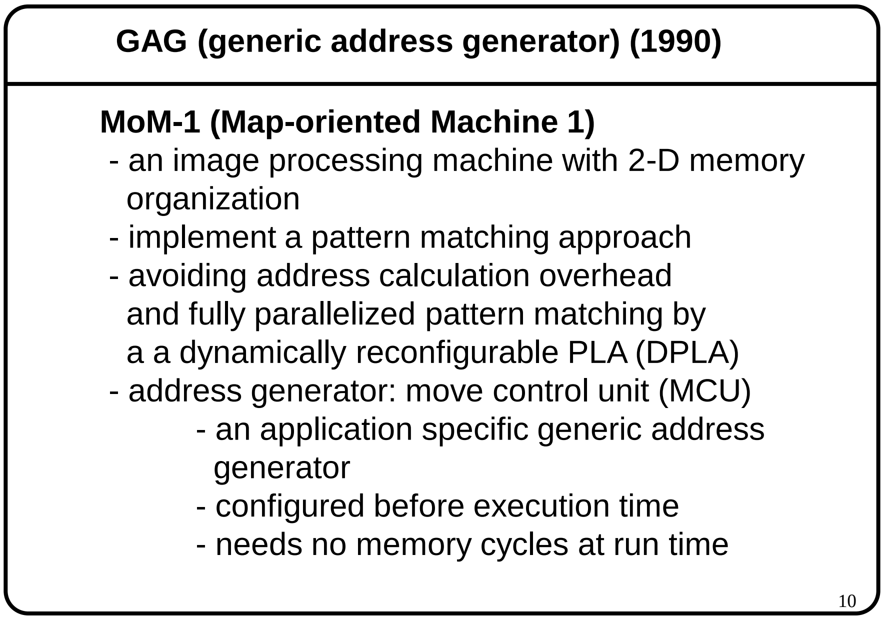
8

The GAG address generator:
How Ingrid' student eplains it

Reiner


DSP Architecture

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing


GAG (generic address generator) (1990)

MoM-1 (Map-oriented Machine 1)
- an image processing machine with 2-D memory organization
- implement a pattern matching approach
- avoiding address calculation overhead and fully parallelized pattern matching by a a dynamically reconfigurable PLA (DPLA)
- address generator: move control unit (MCU)
  - an application specific generic address generator
  - configured before execution time
  - needs no memory cycles at run time

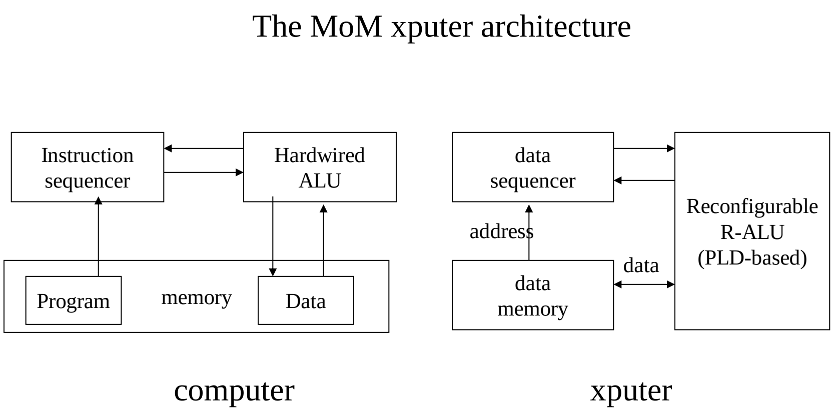
10

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing

The GAG address generator:
How Ingrid' student eplains it

Reiner

GAG (generic address generator) (1990)



11

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing

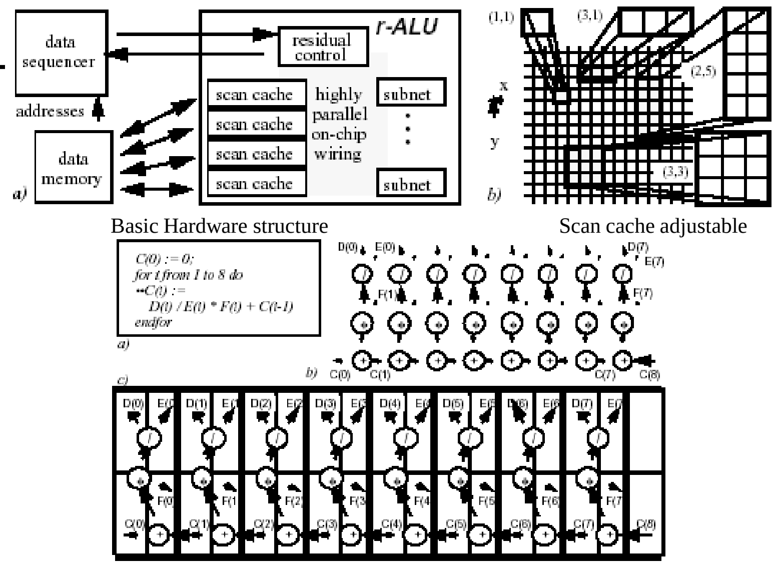


Figure 5: Systolic algorithm example: a) procedural, b) data dependency graph (DG), c) derive data map from DG

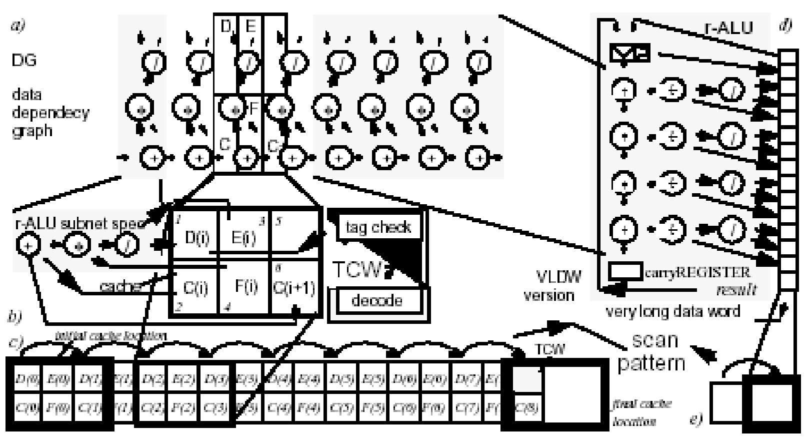
12

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing

The GAG address generator:
How Ingrid' student eplains it

Reiner

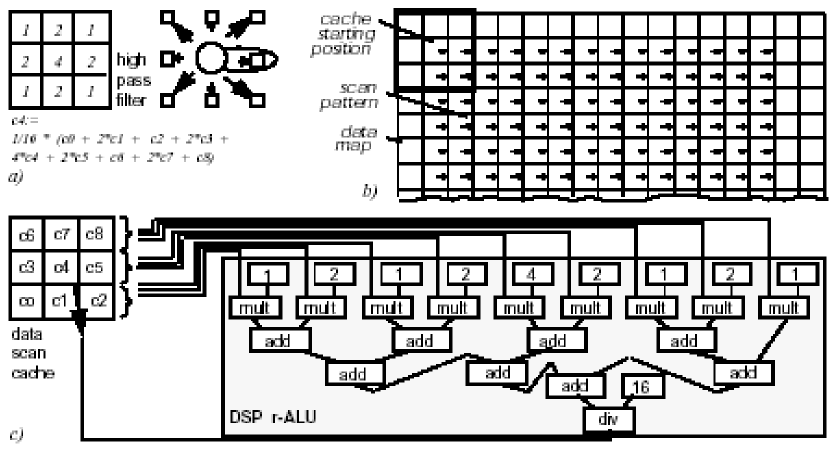
Mapping a Parallel Algorithm



13

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing

2-D Filtering Example



14

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing

The GAG address generator:
How Ingrid' student eplains it

Reiner

GAG of MoM-2 (1991)

- 2 level address generator based on a flexible slider method

- configured by parameters and needs no memory cycles at run time

- Consists:
  1. Jump Generator
  2. Task Manager
  3. Single Step Control Unit(SSCU) - pipeline

GAG of MoM-2

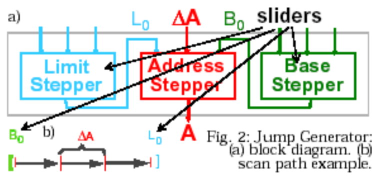


Fig. 2: Jump Generator: (a) block diagram. (b) scan path example.

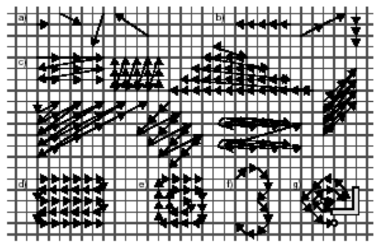
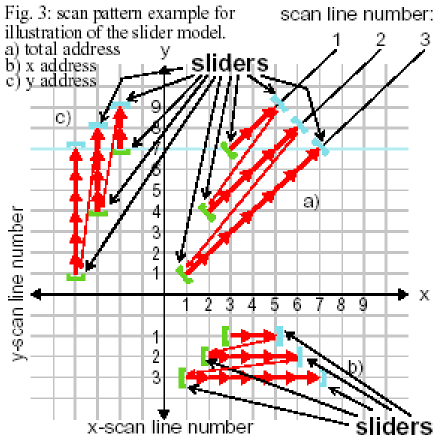


Fig. 1: MoM-3 Scan Pattern Examples: (a) Single Steps, (b) Linear Scans, (c) Video Scans, (d) Zigzag Scan, (e) Spiral Scan, (f) Curve Following (Data Dependent), and (g) Lee Routing (Data Dependent).

Fig. 3: scan pattern example for illustration of the slider model.
a) total address
b) x address
c) y address



The GAG address generator:
How Ingrid' student eplains it

Reiner

GAG of MoM-3 (1993 ~ 1999)

- with Handle Position Generator (HPG) and Memory Address Generator (MAG)
- similar as MoM-2 but improved with multiple (up to 7) access patterns at the same time

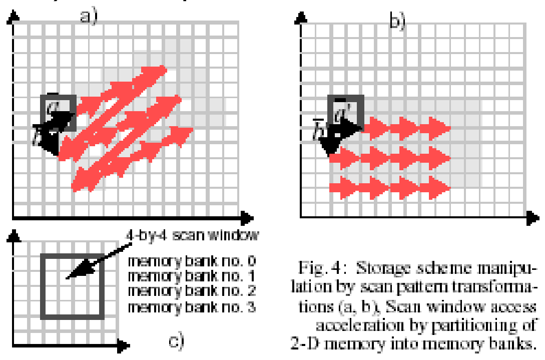
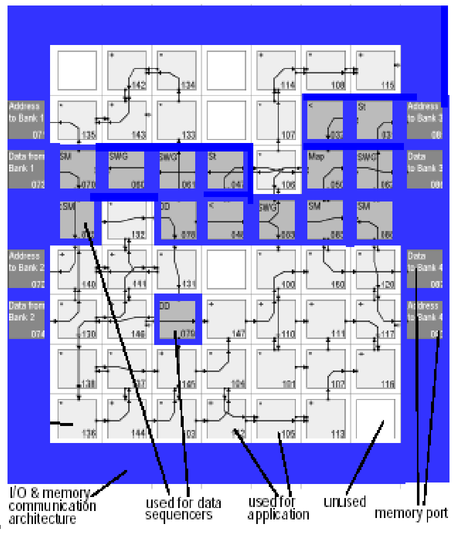


Fig. 4: Storage scheme manipulation by scan pattern transformations (a, b). Scan window access acceleration by partitioning of 2-D memory into memory banks.

17

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing

Mapping Application and Memory Communicatuion

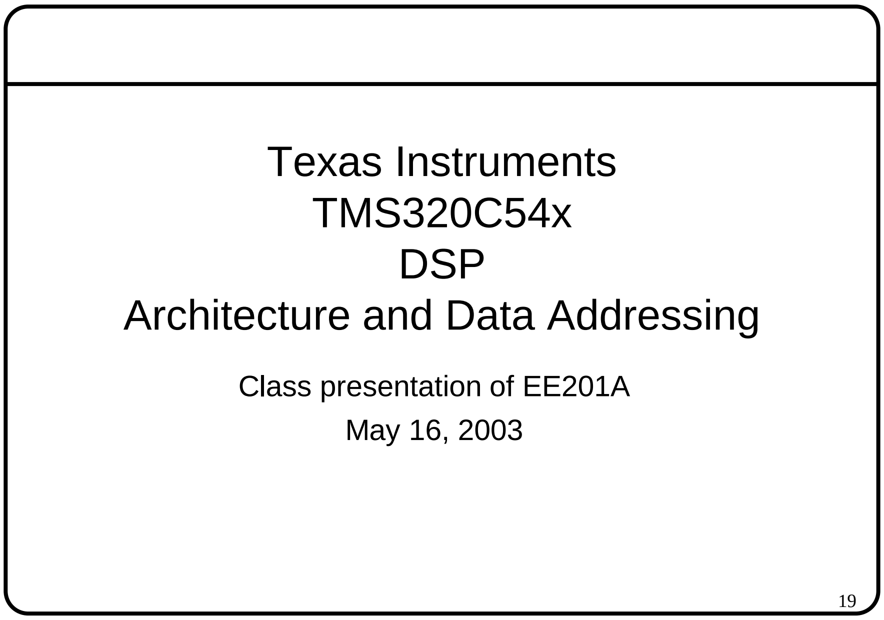


18

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing

The GAG address generator:
How Ingrid' student eplains it

Reiner

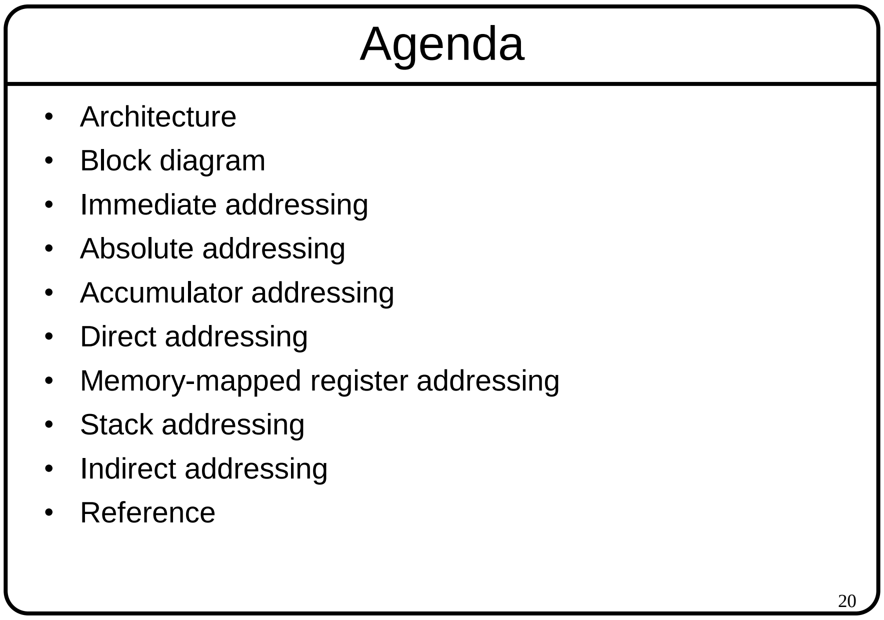


Texas Instruments
TMS320C54x
DSP
Architecture and Data Addressing

Class presentation of EE201A
May 16, 2003

19

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing



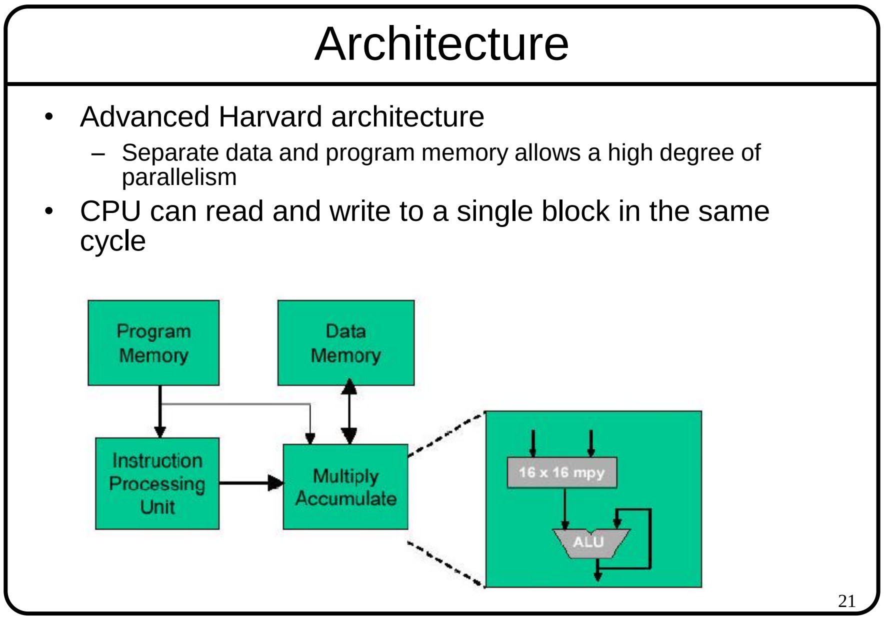
Agenda

- Architecture
- Block diagram
- Immediate addressing
- Absolute addressing
- Accumulator addressing
- Direct addressing
- Memory-mapped register addressing
- Stack addressing
- Indirect addressing
- Reference

20

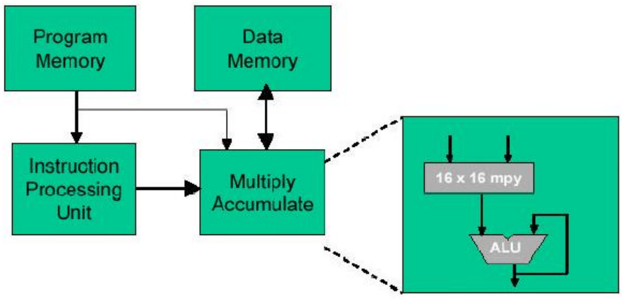
EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing

The GAG address generator:
How Ingrid‘ student eplains it



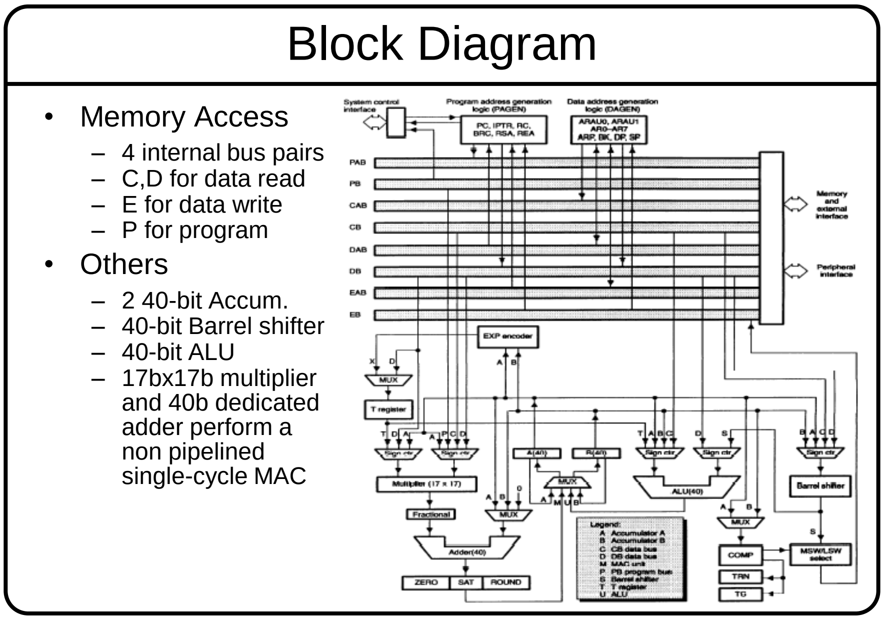
Architecture

- Advanced Harvard architecture
  - Separate data and program memory allows a high degree of parallelism
- CPU can read and write to a single block in the same cycle



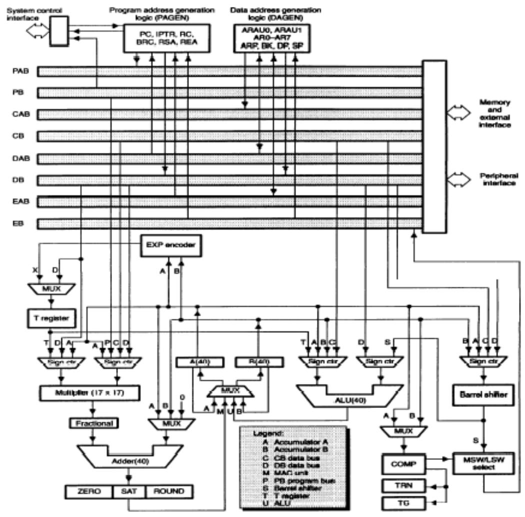
21

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing



Block Diagram

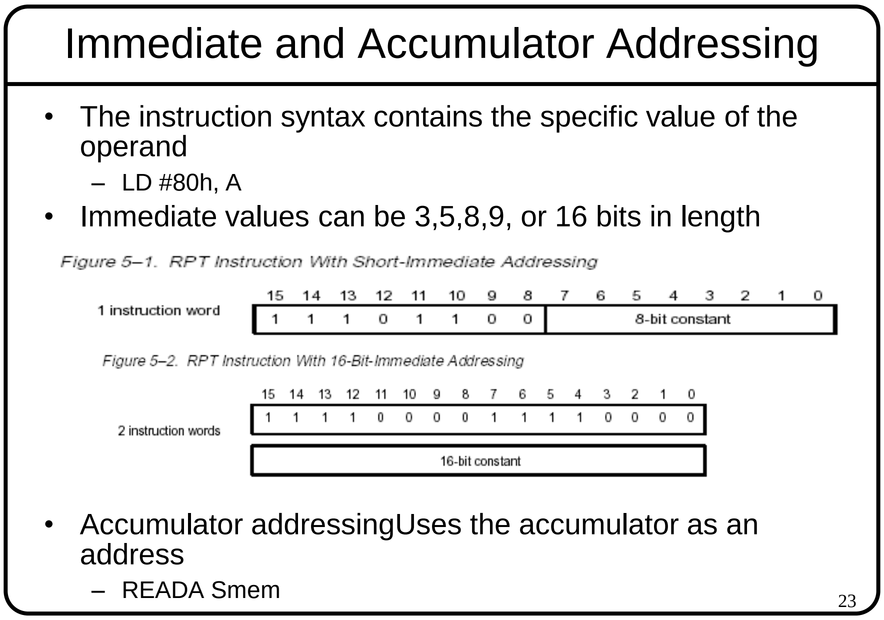
- Memory Access
  - 4 internal bus pairs
  - C,D for data read
  - E for data write
  - P for program
- Others
  - 2 40-bit Accum.
  - 40-bit Barrel shifter
  - 40-bit ALU
  - 17bx17b multiplier and 40b dedicated adder perform a non pipelined single-cycle MAC



EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing

The GAG address generator:
How Ingrid' student eplains it

Reiner



Immediate and Accumulator Addressing

- The instruction syntax contains the specific value of the operand
  - LD #80h, A
- Immediate values can be 3,5,8,9, or 16 bits in length

Figure 5–1. RPT Instruction With Short-Immediate Addressing

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1 instruction word	1	1	1	0	1	1	0	0	8-bit constant							

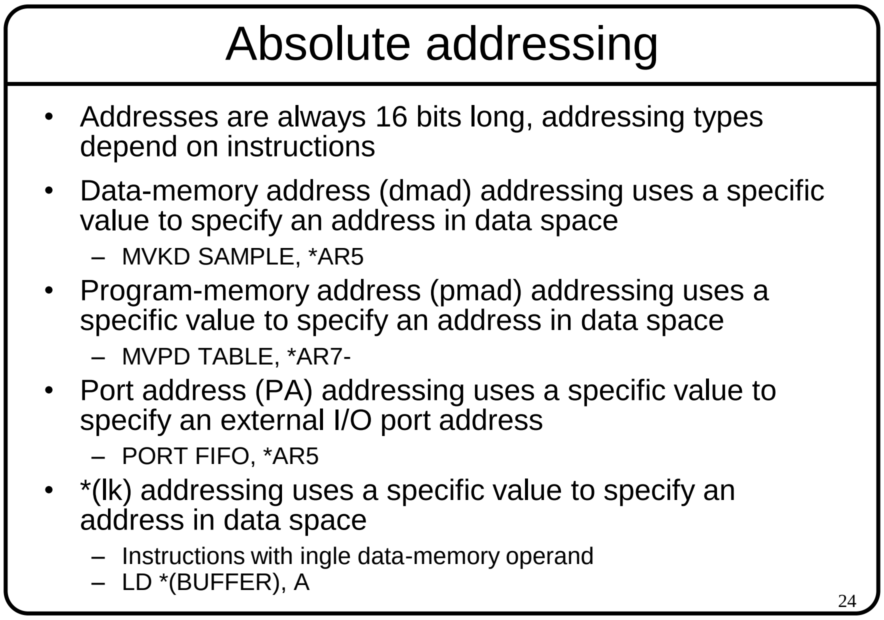
Figure 5–2. RPT Instruction With 16-Bit-Immediate Addressing

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2 instruction words	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
	16-bit constant															

- Accumulator addressingUses the accumulator as an address
  - READA Smem

23

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing



Absolute addressing

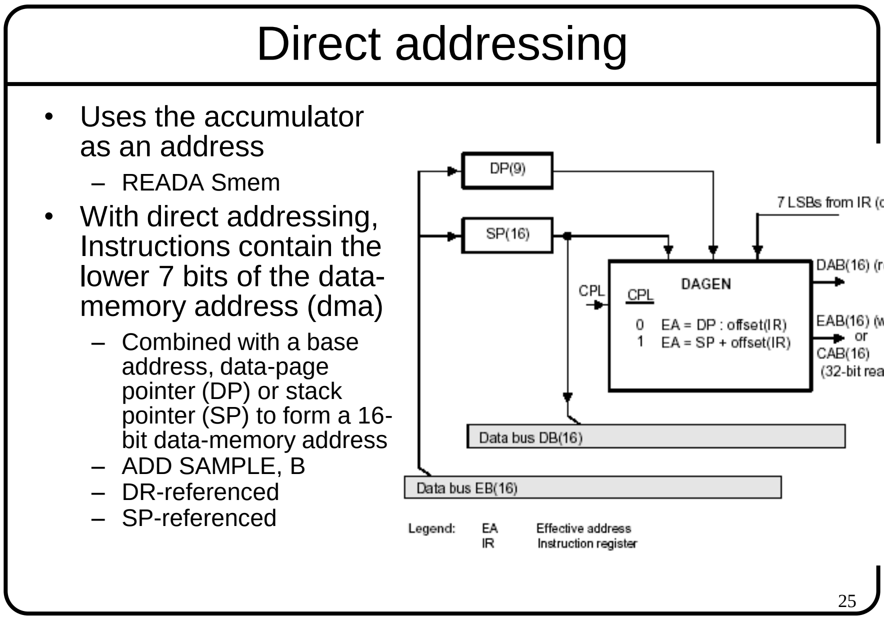
- Addresses are always 16 bits long, addressing types depend on instructions
- Data-memory address (dmad) addressing uses a specific value to specify an address in data space
  - MVKD SAMPLE, *AR5
- Program-memory address (pmad) addressing uses a specific value to specify an address in data space
  - MVPD TABLE, *AR7-
- Port address (PA) addressing uses a specific value to specify an external I/O port address
  - PORT FIFO, *AR5
- *(lk) addressing uses a specific value to specify an address in data space
  - Instructions with ingle data-memory operand
  - LD *(BUFFER), A

24

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing

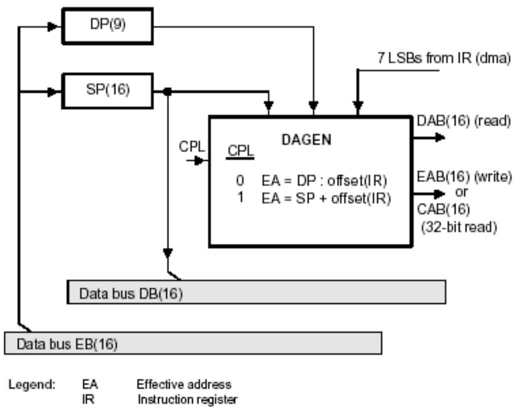
The GAG address generator:
How Ingrid' student eplains it

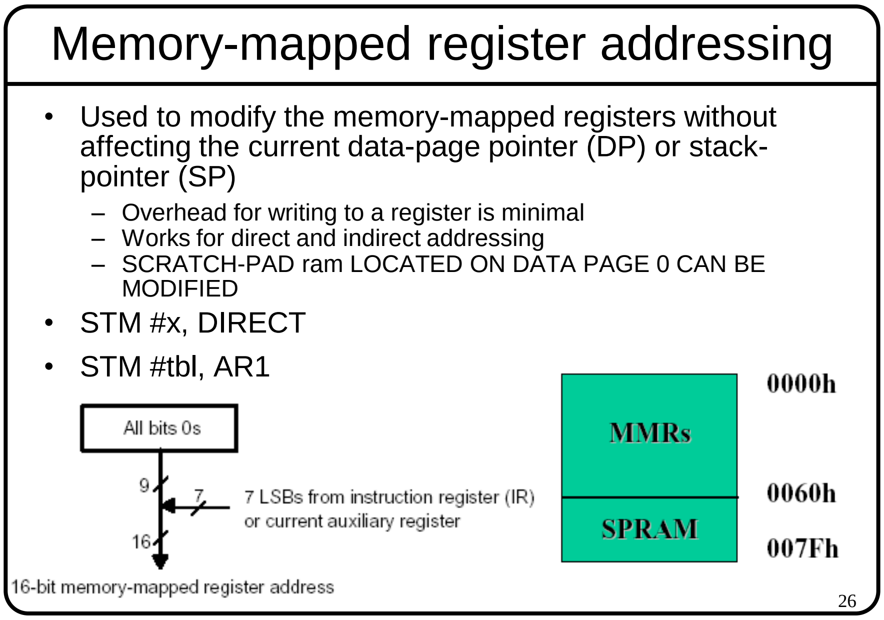
Reiner



Direct addressing

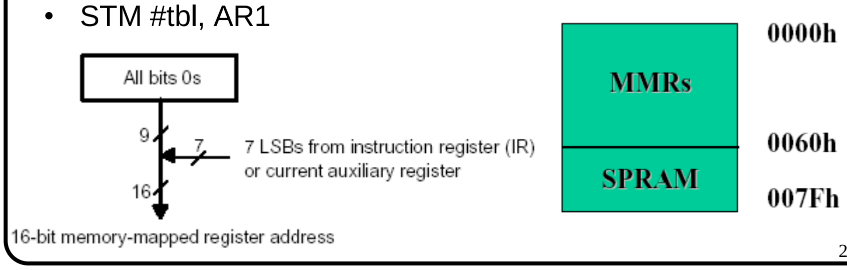
- Uses the accumulator as an address
  - READA Smem
- With direct addressing, Instructions contain the lower 7 bits of the data-memory address (dma)
  - Combined with a base address, data-page pointer (DP) or stack pointer (SP) to form a 16-bit data-memory address
  - ADD SAMPLE, B
  - DR-referenced
  - SP-referenced





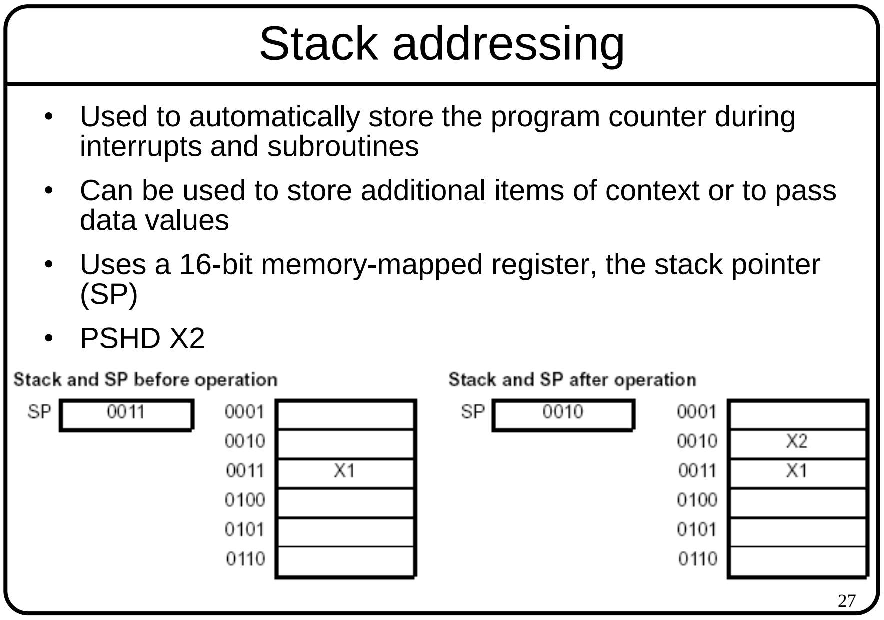
Memory-mapped register addressing

- Used to modify the memory-mapped registers without affecting the current data-page pointer (DP) or stack-pointer (SP)
  - Overhead for writing to a register is minimal
  - Works for direct and indirect addressing
  - SCRATCH-PAD ram LOCATED ON DATA PAGE 0 CAN BE MODIFIED
- STM #x, DIRECT
- STM #tbl, AR1


All bits 0s
9
7
7 LSBs from instruction register (IR) or current auxiliary register
16
16-bit memory-mapped register address
MMRs
SPRAM
0000h
0060h
007Fh

The GAG address generator:
How Ingrid' student eplains it

Reiner



Stack addressing

- Used to automatically store the program counter during interrupts and subroutines
- Can be used to store additional items of context or to pass data values
- Uses a 16-bit memory-mapped register, the stack pointer (SP)
- PSHD X2

Stack and SP before operation

SP 0011

0001	
0010	
0011	X1
0100	
0101	
0110	

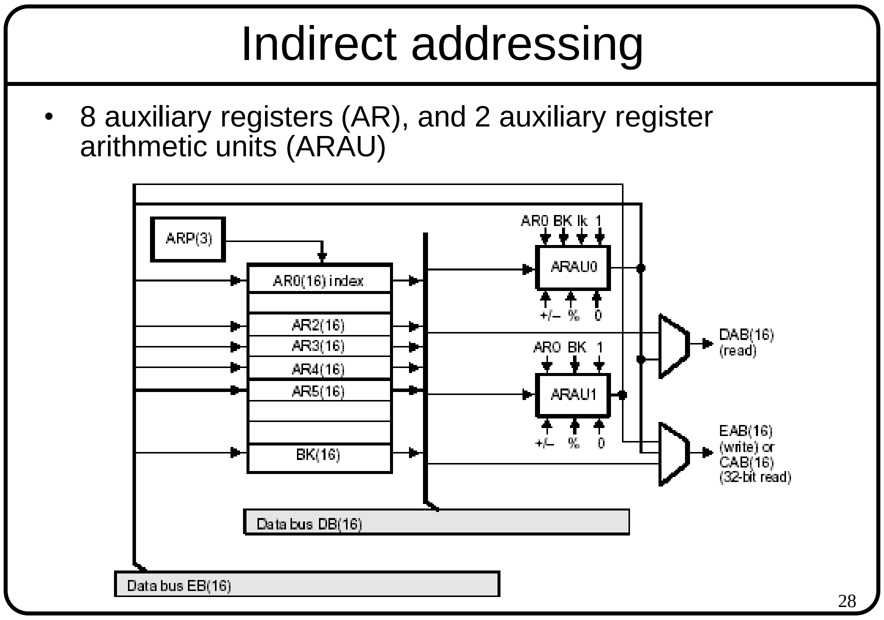
Stack and SP after operation

SP 0010

0001	
0010	X2
0011	X1
0100	
0101	
0110	

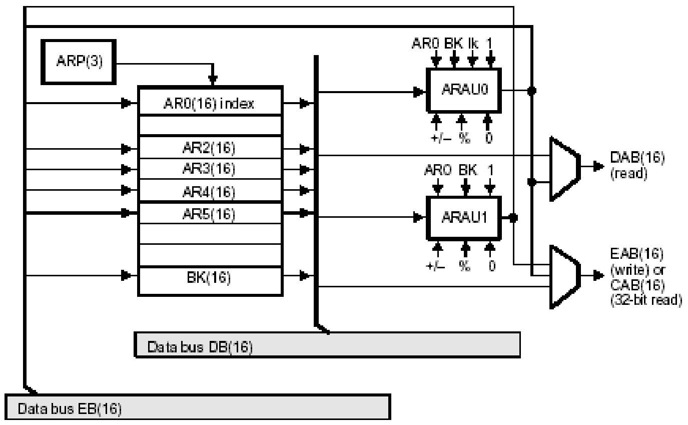
27

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing



Indirect addressing

- 8 auxiliary registers (AR), and 2 auxiliary register arithmetic units (ARAU)

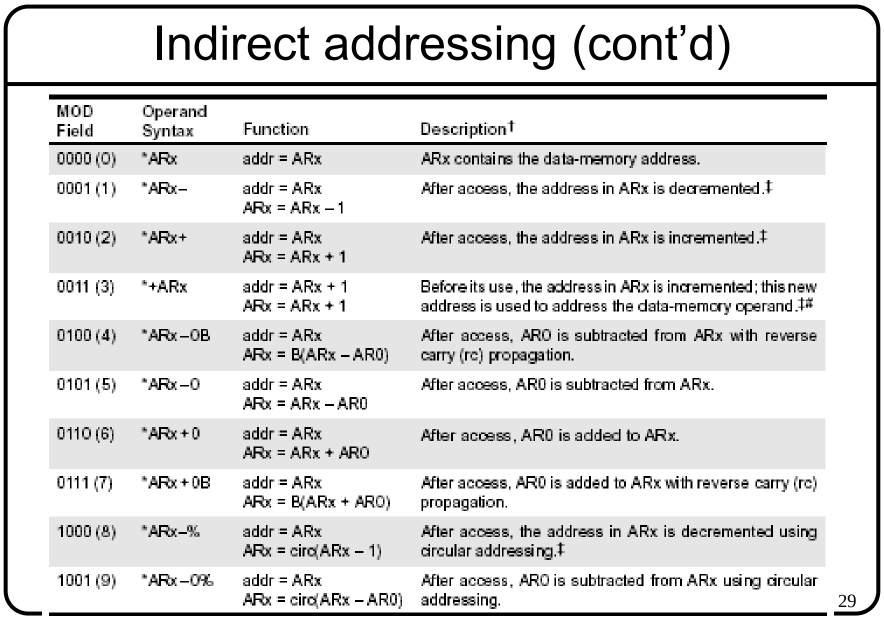


28

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing

The GAG address generator:
How Ingrid' student eplains it

Reiner



Indirect addressing (cont'd)

MOD Field	Operand Syntax	Function	Description†
0000 (0)	*ARx	addr = ARx	ARx contains the data-memory address.

0001 (1)	*ARx–	addr = ARx
		ARx = ARx – 1
			After access, the address in ARx is decremented.‡

0010 (2)	*ARx+	addr = ARx
		ARx = ARx + 1
			After access, the address in ARx is incremented.‡

0011 (3)	*+ARx	addr = ARx + 1
		ARx = ARx + 1
			Before its use, the address in ARx is incremented; this new address is used to address the data-memory operand.‡#

0100 (4)	*ARx–0B	addr = ARx
		ARx = B(ARx – AR0)
			After access, AR0 is subtracted from ARx with reverse carry (rc) propagation.

0101 (5)	*ARx–0	addr = ARx
		ARx = ARx – AR0
			After access, AR0 is subtracted from ARx.

0110 (6)	*ARx+0	addr = ARx
		ARx = ARx + AR0
			After access, AR0 is added to ARx.

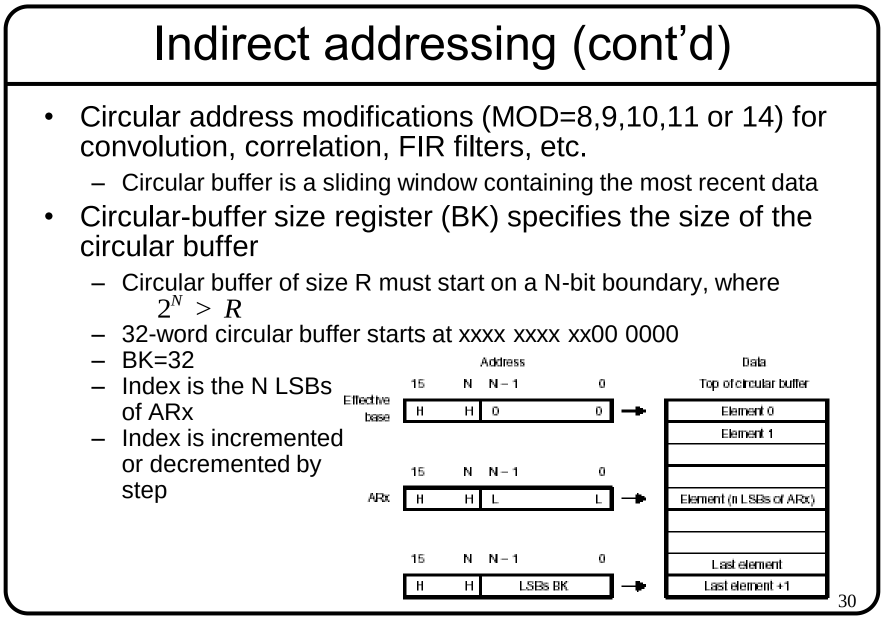
0111 (7)	*ARx+0B	addr = ARx
		ARx = B(ARx + AR0)
			After access, AR0 is added to ARx with reverse carry (rc) propagation.

1000 (8)	*ARx–%	addr = ARx
		ARx = circ(ARx – 1)
			After access, the address in ARx is decremented using circular addressing.‡

1001 (9)	*ARx–0%	addr = ARx
		ARx = circ(ARx – AR0)
			After access, AR0 is subtracted from ARx using circular addressing.

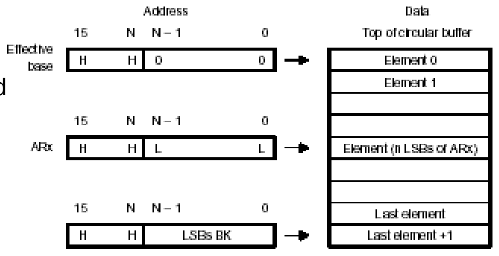
29

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing



Indirect addressing (cont'd)

- Circular address modifications (MOD=8,9,10,11 or 14) for convolution, correlation, FIR filters, etc.
  – Circular buffer is a sliding window containing the most recent data
- Circular-buffer size register (BK) specifies the size of the circular buffer
  – Circular buffer of size R must start on a N-bit boundary, where $2^N > R$
  – 32-word circular buffer starts at xxxx xxxx xx00 0000
  – BK=32
  – Index is the N LSBs of ARx
  – Index is incremented or decremented by step



30

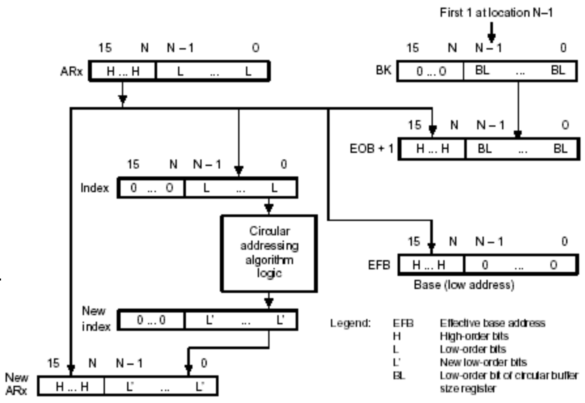
EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing

The GAG address generator:
How Ingrid' student eplains it

Reiner

Indirect addressing (cont'd)

If $0 \le$ index + step $<$ BK:
  index = index + step.
Else if index + step $\ge$ BK:
  index = index + step – BK.
Else if index + step $<$ 0:
  index = index + step + BK.



31

Indirect addressing (cont'd)

- Bit-Reversed Address Modifications (MOD=4 or 7)
  - Enhances execution speed and program memory for FFT algorithms that use a variety of radixes
- Assume FFT size is 2^N , then AR0= 2^{N-1}
  - An ARx points to the physical location of a data value

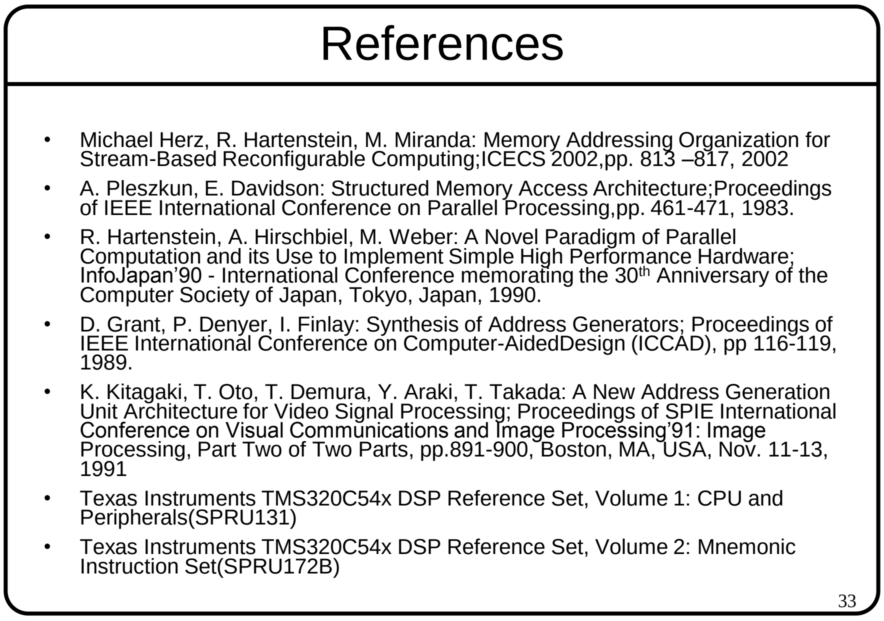
```
*AR2+0B   ;AR2   =   0110 0000    (0th value)
*AR2+0B   ,AR2   =   0110 1000    (1st value)
*AR2+0B   ;AR2   =   0110 0100    (2nd value)
*AR2+0B   ;AR2   =   0110 1100    (3rd value)
*AR2+0B   ;AR2   =   0110 0010    (4th value)
*AR2+0B   ;AR2   =   0110 1010    (5th value)
*AR2+0B   ;AR2   =   0110 0110    (6th value)
*AR2+0B   ;AR2   =   0110 1110    (7th value)
```

Bit-Reversed Pattern	Bit-Reversed Step
0000	0
1000	8
0100	4
1100	12
0010	2
1010	10
0110	6
1110	14

32

The GAG address generator:
How Ingrid' student eplains it

Reiner



References

- Michael Herz, R. Hartenstein, M. Miranda: Memory Addressing Organization for Stream-Based Reconfigurable Computing;ICECS 2002,pp. 813 –817, 2002
- A. Pleszkun, E. Davidson: Structured Memory Access Architecture;Proceedings of IEEE International Conference on Parallel Processing,pp. 461-471, 1983.
- R. Hartenstein, A. Hirschbiel, M. Weber: A Novel Paradigm of Parallel Computation and its Use to Implement Simple High Performance Hardware; InfoJapan'90 - International Conference memorating the 30th Anniversary of the Computer Society of Japan, Tokyo, Japan, 1990.
- D. Grant, P. Denyer, I. Finlay: Synthesis of Address Generators; Proceedings of IEEE International Conference on Computer-AidedDesign (ICCAD), pp 116-119, 1989.
- K. Kitagaki, T. Oto, T. Demura, Y. Araki, T. Takada: A New Address Generation Unit Architecture for Video Signal Processing; Proceedings of SPIE International Conference on Visual Communications and Image Processing'91: Image Processing, Part Two of Two Parts, pp.891-900, Boston, MA, USA, Nov. 11-13, 1991
- Texas Instruments TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals(SPRU131)
- Texas Instruments TMS320C54x DSP Reference Set, Volume 2: Mnemonic Instruction Set(SPRU172B)

EE201A, Spring 2003, Yung-Szu Tu, Chun-Ching, UCLA - Memory Addressing

The GAG address generator:
How Ingrid' student eplains it