

Trade-off Considerations and Performance of Intel's MMX™ Technology

**Uri Weiser
Intel Corporation
Israel Design Center**

August 20, 1996

Agenda

- **The Opportunity**
- **Definition Consideration**
- **MMX™ Technology**
- **Performance/Example**
- **Conclusions**

The Opportunity

- **Emergence of new applications**
 - **Multimedia**
 - **Communication**
- **The need for performance**
- **Utilization of existing hardware**
 - **Datapath**
 - **Registers**
 - **Internal buses**

Evolution of the PC

- **The trend: Multimedia and communications applications are driving the market**
 - Video & audio compression (DVD, MPEG2, AC3), games (3D graphics), speech recognition, voice compression, image processing, video conferencing (POTS and LAN)

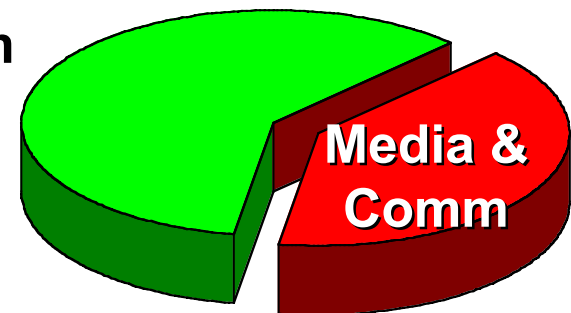
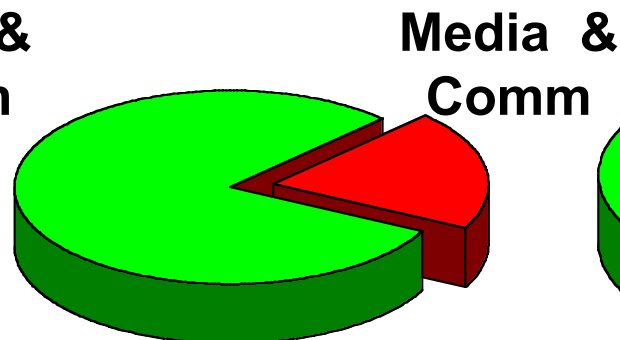
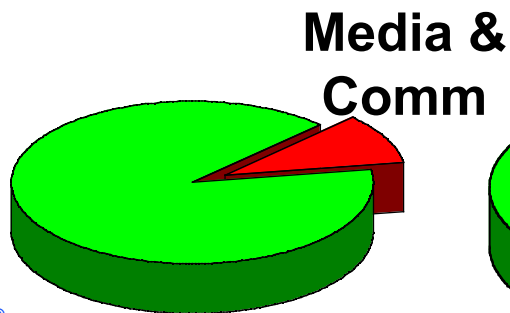
The Home PC

- Same as office

The Multimedia PC

- Audio
- CD-ROM
- Graphics accelerator
- Modem

Next ?



Characteristics of Multimedia and Communication Apps/Algorithms

- **Multimedia and communication applications built from basic algorithms “glued” together**
- **Large degree of commonality across the diverse algorithms**
 - **Computation intensive**
 - **Data streaming**
 - **Small data types**
 - **Potential data parallelism**

Definition Consideration

- **Full compatibility with existing Intel architecture software model**
- **Significant performance benefit**
- **General and flexible/not specific**
- **Minimal die size impact and design complexity**

Compatibility

Requirements:

- **Map into existing Intel Architecture**
 - No new machine state
 - No new events
 - Availability of unused Op Code space

Approach:

- **Use of FP registers structure**
(80/64 bit vs. 32 bit integer registers)
- **No new exceptions**
- **Alias of FP OS handling mechanisms**

Generality/Extensibility

- **Define Atomic operations**
 - Arithmetic → add, sub, shift, mul, compare
 - Logic → and, andnot, or, xor
 - Conversion
 - Exception: Muladd
- **Straight forward migration into Intel's future processors**

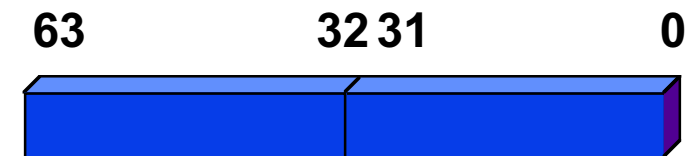
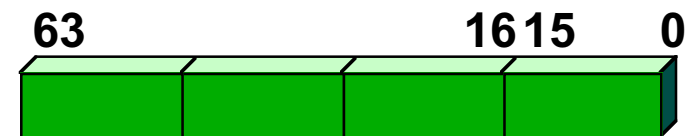
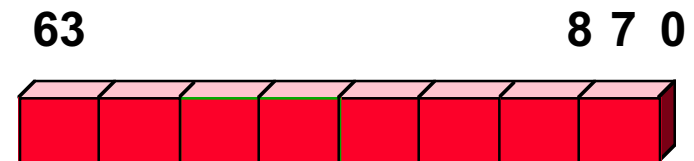
Intel's MMX™ Technology

- **57 new Instructions**
 - **Single Instruction Multiple Data Architecture technique (SIMD)**
 - **Fixed point integer**
- **Map into 8 FP registers/direct access**
- **No new exceptions**
- **Low implementation complexity**

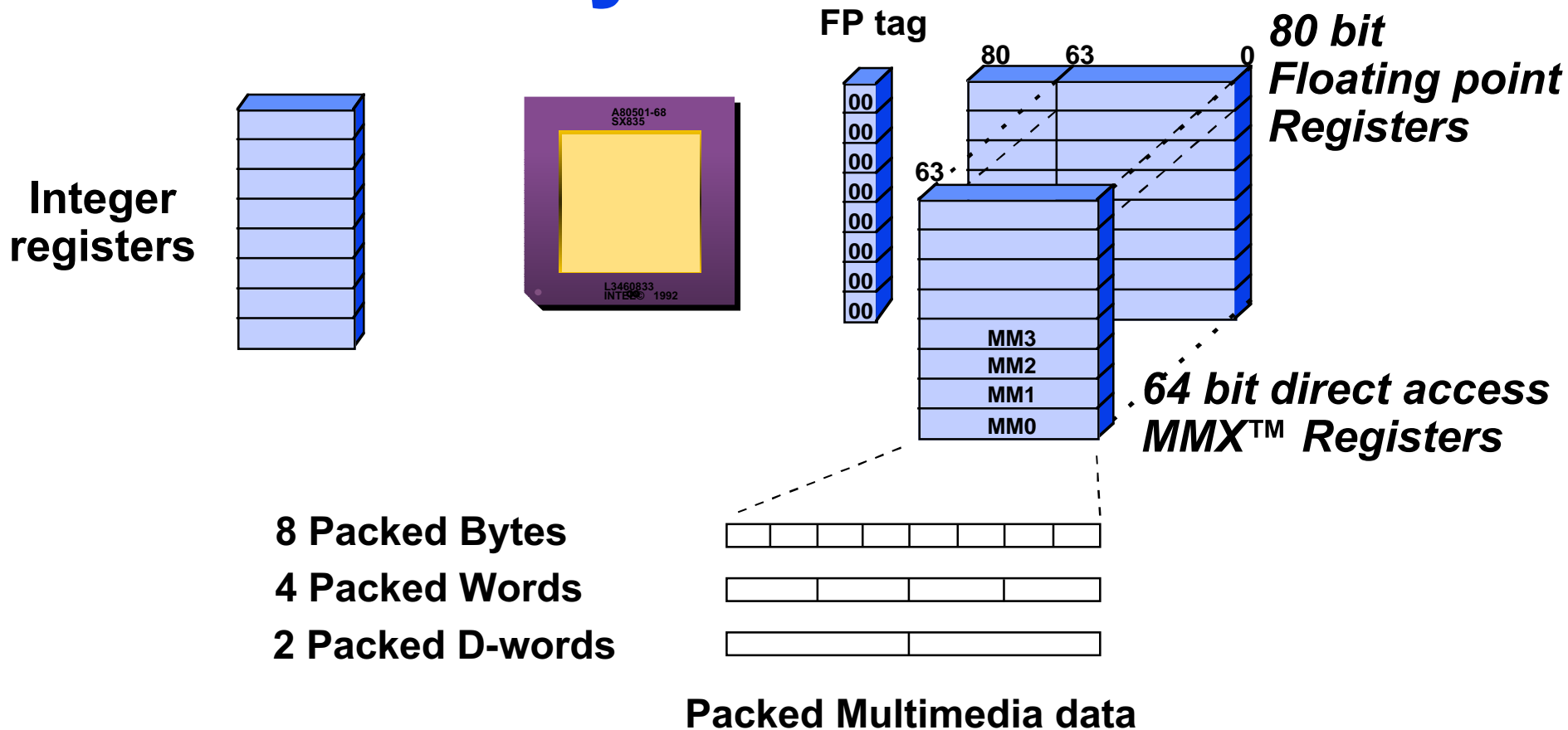
The Most Significant Enhancement to Intel Architecture Since the i386™ Processor

Data Types

- **Packed bytes**
 - Mainly for graphics and video
- **Packed words**
 - Used mainly for audio and comm.
- **Packed doublewords**
 - General purpose use
- **Quadword**
 - Bitwise operations and Data alignment



MMX™ Architecture Summary



A Compatible Extension Architecture

Sample MMX™ Technology Operations

Saturating Arithmetic

a3	a2	a1	FFFFh
+	+	+	+
b3	b2	b1	8000h
<hr/>			
a3+b3	a2+b2	a1+b1	FFFFh

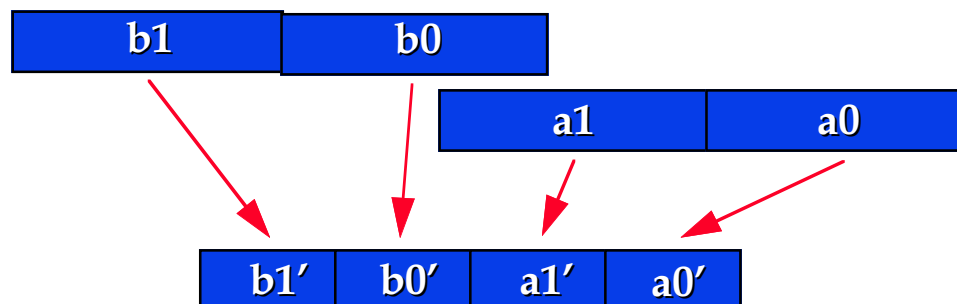
Parallel Compares

23	45	16	34
gt?	gt?	gt?	gt?
31	7	16	67
<hr/>			
0000h	FFFFh	0000h	0000h

16b x 16b => 32b Multiply Add

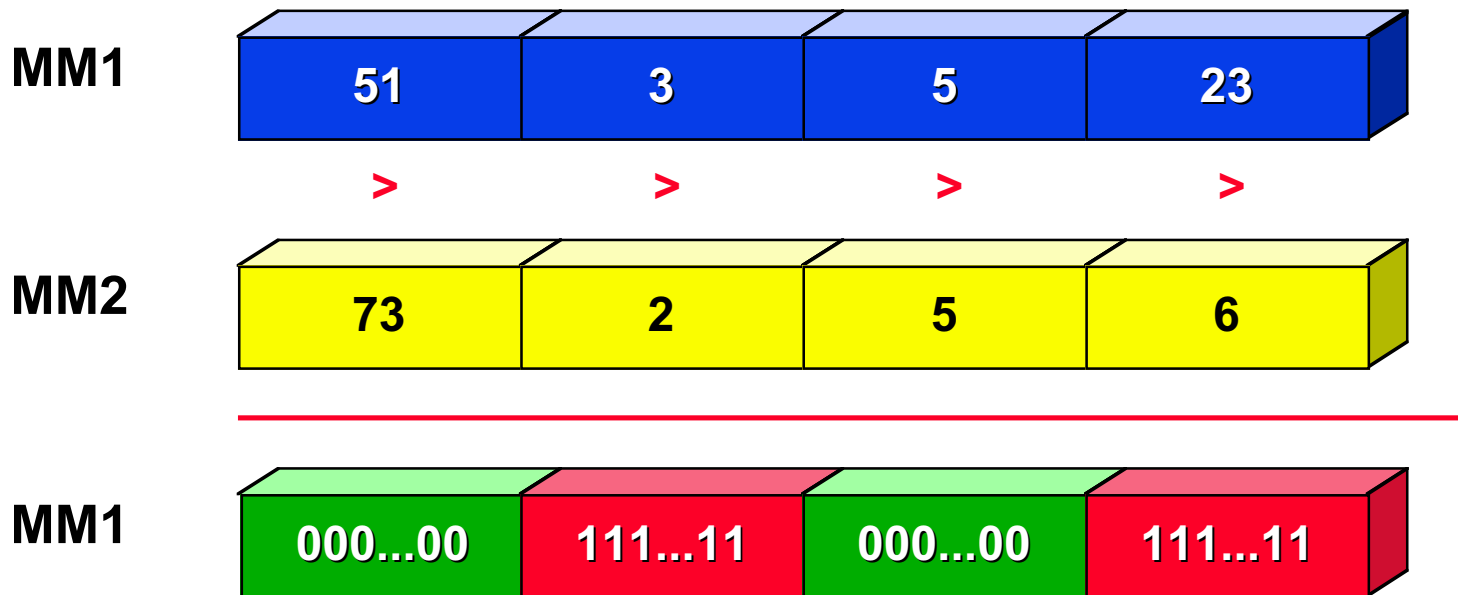
a3	a2	a1	a0
*	*	*	*
b3	b2	b1	b0
<hr/>			
a3*b3+a2*b2		a1*b1+a0*b0	

Data Conversion



What Is A Parallel Compare?

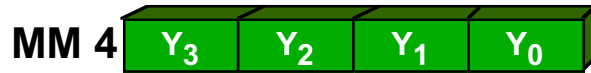
- No flags to store multiple results
- Result is a mask



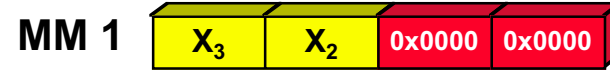
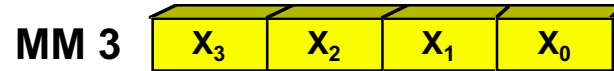
Conditional Selection



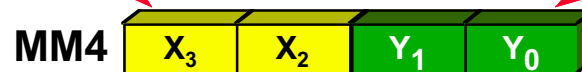
pand MM4, MM1



pandn MM1, MM3



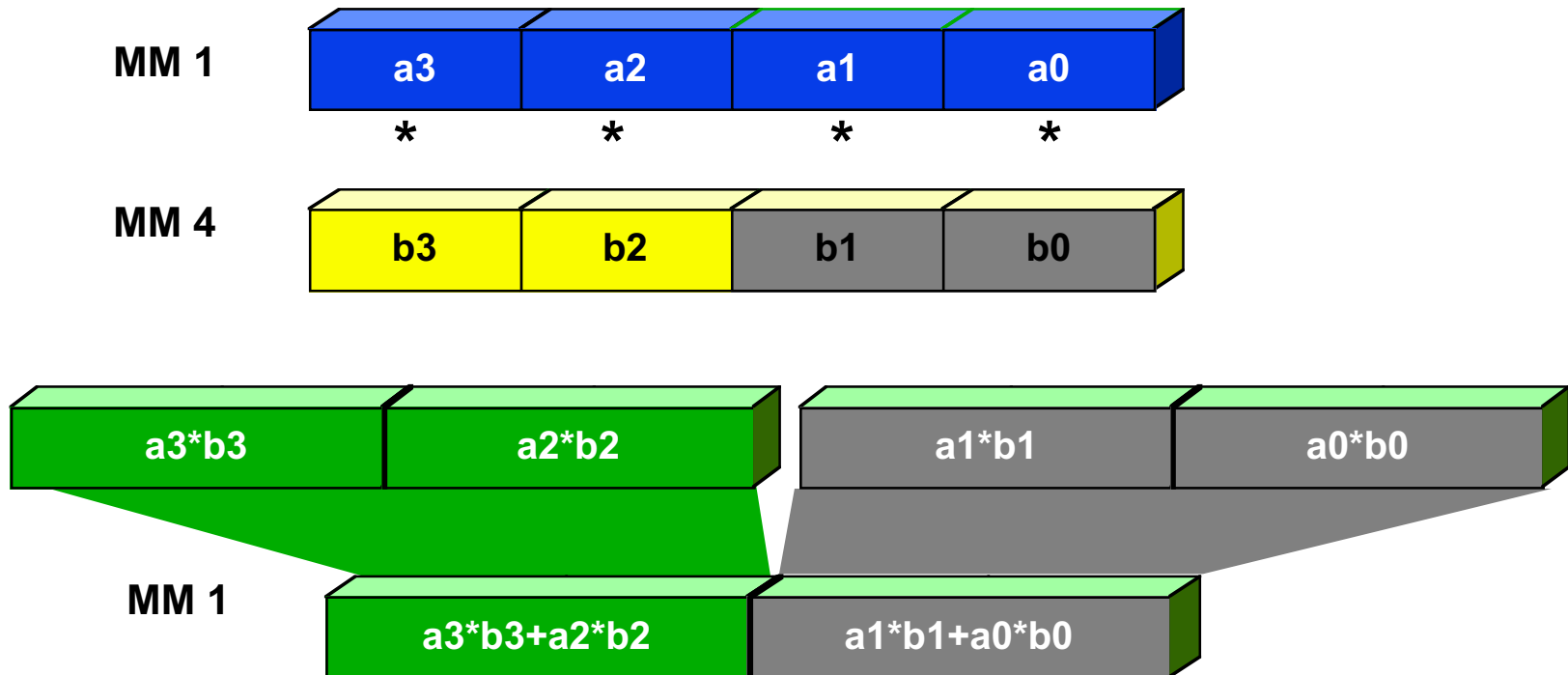
por MM4, MM1



PMADD

`pmaddwd` MM1, MM4

packed **m**ultiply and **a**dd 4 words to 2 **d**oublewords

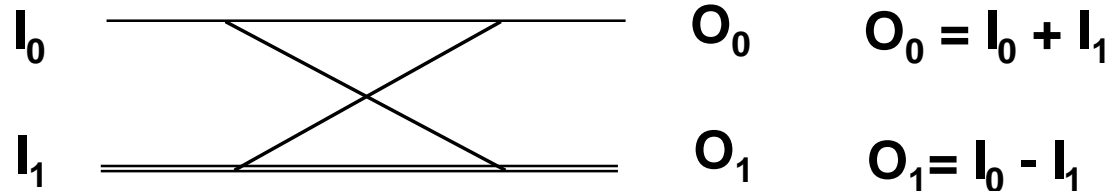


MMX™ Technology Code Example

Inverse Discrete Cosine Transform (IDCT) Scalar vs. MMX™ Technology

Used in Video compression/decompression standards*

**Basic Operation:
A butterfly**



Scalar

```
mov    eax, [edi] /* load 1st value
mov    ebx, [edx] /* load 2nd value
mov    eax, esi   /* Copy 1st value
sub    eax, ebx   /* O1 = I0 - I1
add    esi, ebx   /* O0 = I0 + I1
```

MMX Technology

```
Movq   mm4, [edi] /* load 1st 4 values
Movq   mm7, [edx] /* load 2nd 4 values
Movq   %mm4,%mm0 /* Copy 1st values
Psubw  %mm7,%mm4 /* O1[0-3] = I0[0-3] - I1[0-3]
Paddw  %mm0,%mm7 /* O0[0-3] = I0[0-3] + I1[0-3]
```

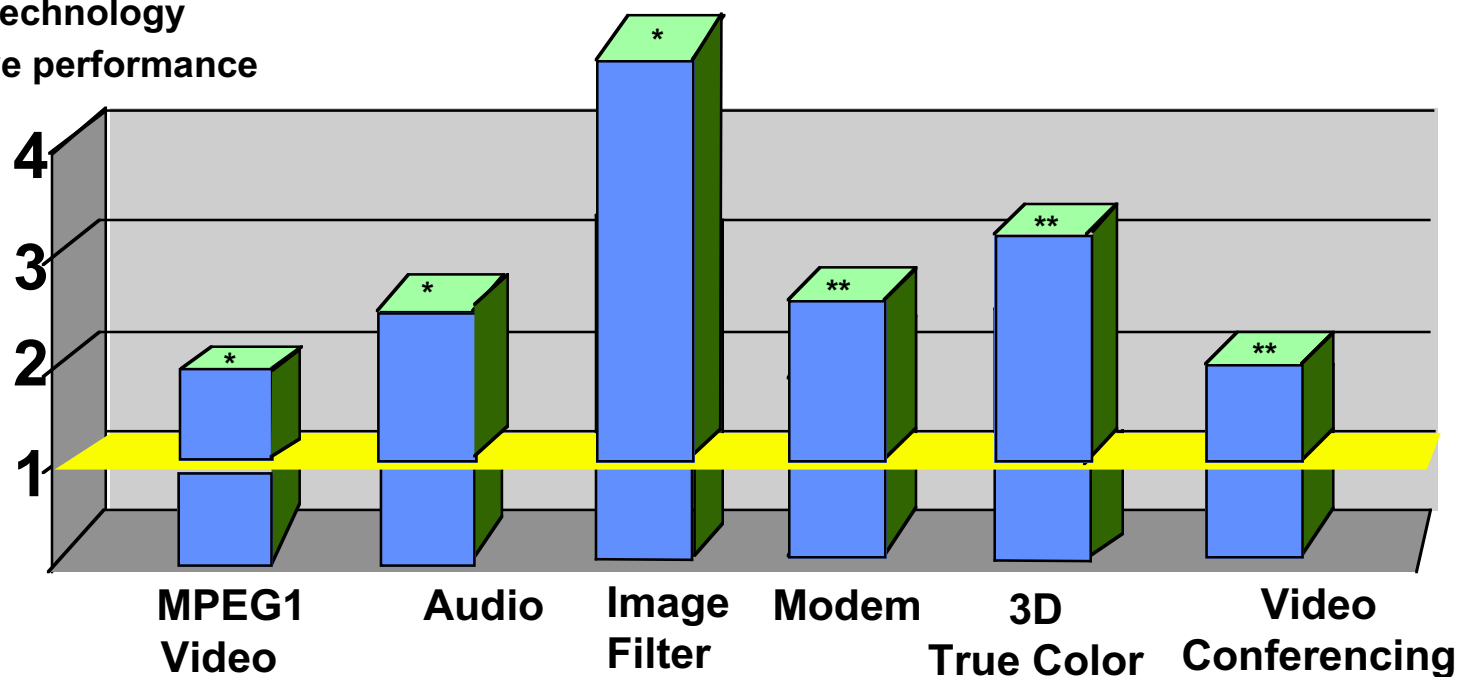
**Same Operation, Same Amount of Instructions
MMX Technology 4X Faster**



* LLM algorithm - Practical fast 1-D DCT - 1989 - assume data elements are 16 bits,

MMX™ Technology Performance

MMX Technology
Relative performance



* Measured: Components of Intel's Media Benchmark
** Estimated. Based on inner loops and algorithm analysis

Conclusions

- **Implementation shows full compatibility with existing OS and applications**
- **Simple definition → clean implementation**
- **Performance improvement of multimedia application 1.5- 5X**