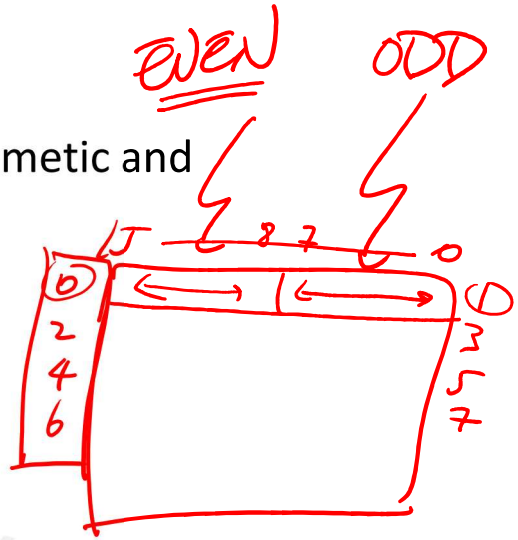
ECE 242
Digital Computers and Computing

Lecture 4. Intro to MC68K – Part 1
Spring 2021

Won-Jae Yi

Architecture Features

- Register Set
  - Data, address and special purpose registers for on-chip storage
- Instruction Set
  - 56 basic instructions for data movement, arithmetic and logical operations, and program control
- Addressing Modes
  - 14 modes to designate operands in memory
- Machine Language
  - Defines the binary format of the instruction
- Memory Organization
  - Specifies the format of instructions and operands in memory



Instruction Set Design

- MC68K Family instructions categories as follows:
  - Data movement
  - Arithmetic operations
  - Logical operations
  - Bit manipulation and shift operations
  - Program control
  - System control
- Evident goal to reduce the gap between HLL and CPU instructions
- This approach gave rise to notion of CISC (Complex Instruction Set Computer)

this is what we do in HLL.

thus, can be translated to CPU inst. easily.

flexibility

↳ let's have many inst. available

Reduced Instruction Set Computer

- Fixed instruction size, one-cycle execution
  - Same length of each instruction, simplifying instruction fetching and decoding
- Load and store architecture
  - Fetch operands from memory → load to registers → arithmetic/other operations → store back to memory
  - Each requires one instruction
- Large number of registers (32 or more)
  - Load and store architecture
- Small instruction set
  - Difficult assembly language programming
- Compiler optimization necessary

Complex Instruction Set Computer

- Variable instruction size and execution times
  - Desire to minimize the use of memory for instructions
- Memory operation possible
  - Some instructions perform the memory as well as perform arithmetic operations
- Limited register set
  - Memory accesses are sometimes necessary to store data temporarily
- Large and complete instruction set
  - Relatively simple assembly language programming
- No special compiler requirements

MC68K Registers

- Registers are just as the memory, used to store instructions and data associated with a program where it is storage elements of the CPU, holding information needed for the instruction currently being processed.
- Programmable registers of MC68K
  - **General-purpose registers**
    - 8 Data registers
    - 9 Address registers (A0-A6, A7 as user or supervisor system stack pointer)
  - **Program counter**
  - **Status register or condition code register**

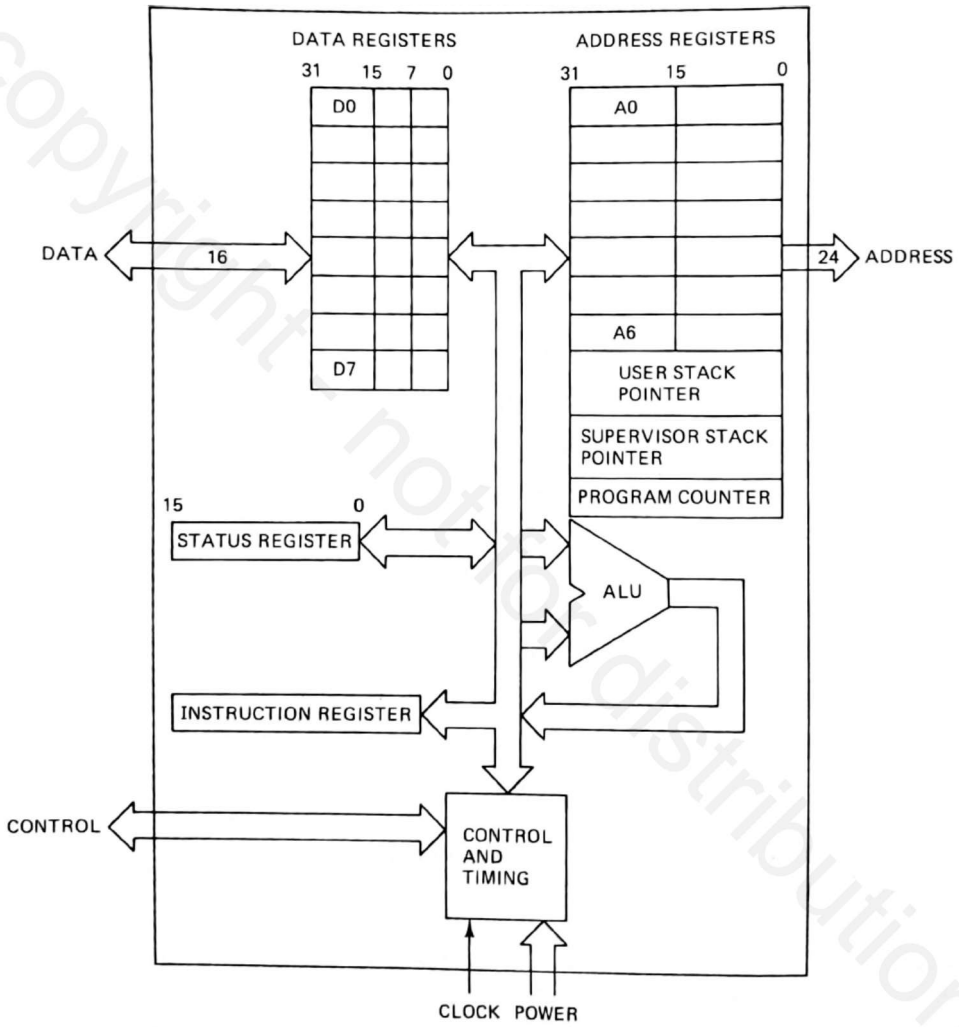


Figure 4.1 The MC68000 register set and transfer paths.

MC68K Registers

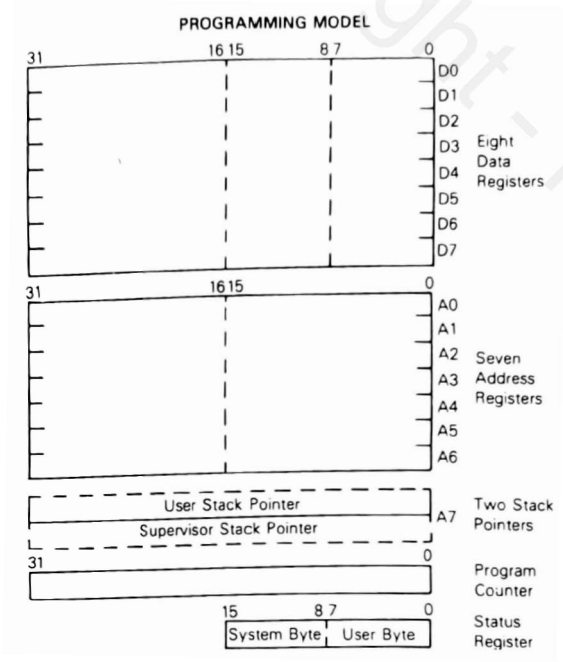


Table 4.4 Register Usage and Symbolic Notation

Register	Symbolic Notation	Usage
Data	D0	Accumulator
	D1	Buffer register
	•	Index register
	•	Temporary storage
	•	
	D7	
Address	A0	Indirect addressing
	A1	Stack pointer
	•	Index register
	•	
	•	
	A6	
System stack pointer	A7 or SP	
User SP	A7 or USP	User subroutine calls
Supervisor SP	A7 or SSP	Interrupt processing or subroutine calls (supervisor mode)
Program counter	PC	Instruction addressing
Status register	SR	System status, condition codes

Data Registers

- D0 to D7 (8 data registers)
- Three lengths are possible with data
  - A byte operand (Dn)[7:0]
  - A word operand (Dn)[15:0]
  - A longword operand (Dn)[31:0]
- Processor instructions must indicate the operand length when referencing a data register
  - Only the corresponding bits of the specified register are modified by that instruction
- As MC68K uses 16 data signal lines
  - Can perform an 8-bit or 16-bit in a single transfer
  - Can perform a 32-bit value in two transfers of 16 bits each
- Data register can be used as an *index* register
  - Its contents are added to the value in an address register to form an address of an operand

Address Registers

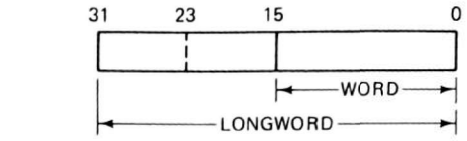
- 9 address registers in MC68K
  - Accepts word or longword values only
  - A0 to A6: shared by programs in either supervisor or the user mode
  - A7: referenced as user stack pointer (USP) or system stack pointer (SSP)
- Holds the address of an operand in memory
- An address register may range up to 2^{32}
  - Only the lower 24 bits allowed via address/control lines in MC68K

Address Registers

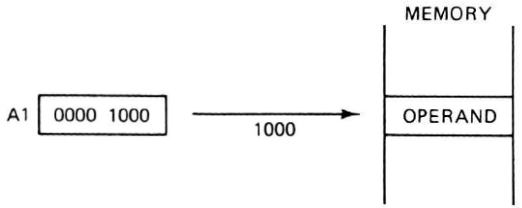
- Use address registers for Private Stacks
  - A **stack** consists of a set of contiguous memory locations addressed by a register designated as a **stack pointer**
  - Items stored on the stack are retrieved in LIFO (last-in-first-out) manner
  - Stack pointer indicates the top of the stack
    - SSP or USP holds an address that *points* to the top of the stack
  - Data stored on the stack is said to be ***pushed*** on the stack
  - Data retrieved from the stack is said to be ***pop*** or ***pull***.
- Use address registers as an Index Register
  - Index value is added to the contents of another address register to compute an effective address
  - Similar to the usage of a data register

Address Registers

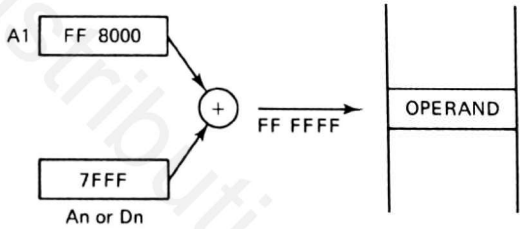
- An indirect memory reference is denoted as (A1) in assembly language, where the effective address of the operand is designated as the contents of A1
- Indexed addressing allows the sum of two values to be used to determine the operand location
- Maximum 24-bit address of a byte location in memory is FF $FFFF_{16}$



(a) Address Register



(b) Indirect Memory Reference

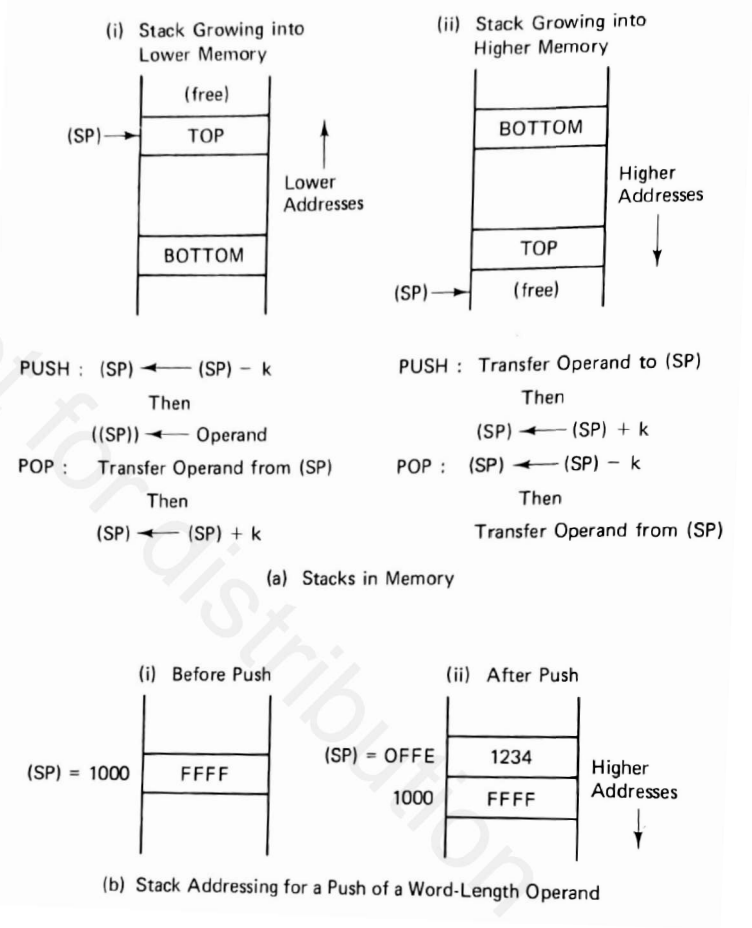


(c) Indexed Memory Reference

Note: All values are in hexadecimal.

Address Registers

- SP : Stack Pointer
- k: 1 (byte), 2(word), 4(longword)
- Predecrement mode would subtract k from (SP) before use
- Postincrement mode would add k to (SP) after use



System Stack Pointers

- MC68K uses stack when:
  - A subroutine call is made by a program
  - An *exception* occurs during system operations (e.g., traps, interrupts, several error conditions)
- When any of these events occur, control is passed to the instructions associated with the subroutine, trap or interrupt until its specific task is completed. Then, control is normally returned to the next instruction

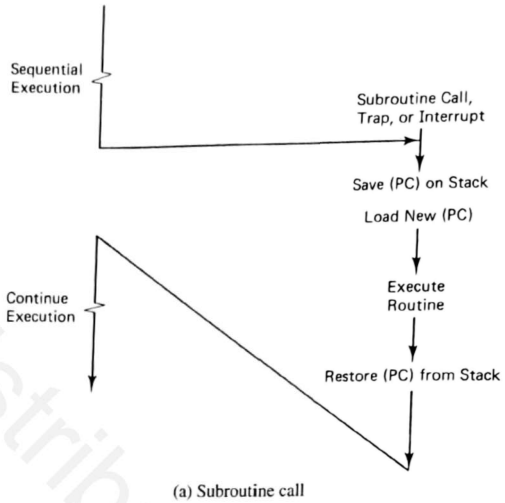
- During this process:
  - A subroutine call: value of (PC) saved/restored to/from the stack
  - An *exception*: value of (PC) and value of (SR) <- status register

System Stack Pointers

- MC68K separates user stack area and supervisor stack area
- User mode's system stack is user stack area, can be accessed by User Stack Pointer (USP)
- Supervisor's system stack is supervisor stack area, can be accessed by Supervisor Stack Pointer (SSP)
- Any time MC68K executes a program, only one system stack is being used (A7 register)

System Stack Pointers – Subroutine

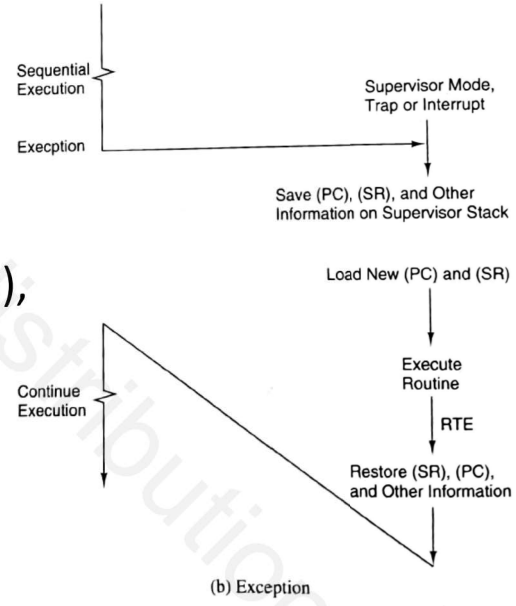
- Subroutine Calls are executed by Jump to Subroutine (JSR) or Branch to Subroutine (BSR) instructions
- Program counter (PC, 32-bit) saved to system stack
- New PC loaded from JSR or BSR instructions
- When subroutine completes, return to original locations by calling Return from Subroutine (RTS)
  - Reloads saved PC from the system stack
  - Saved PC indicates the address following JSR or BSR instruction



(a) Subroutine call

System Stack Pointers – Exception

- CPU automatically controls exception handling routine
- Changes mode to supervisor mode
- (PC) and (SR) saved on the supervisor stack
- After the exception routine complete execution, the last instruction RTE should be used to restore (PC) and (SR), and return to original program

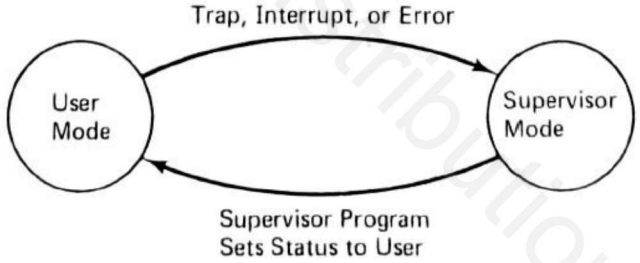


(b) Exception

Supervisor vs User Mode

- One of privilege concerning the control of the CPU or external device
- User mode may be restricted to a certain instruction, a certain memory area, a certain external device
- Supervisor can override anything including user's program and memory
- Transition between modes:

Transition between modes.



Supervisor vs User Mode

Table 4.5 Supervisor Versus User Mode

	Supervisor	User
Register usage	D0–D7, A0–A6, PC, SSP	D0–D7, A0–A6, PC, CCR
Stack pointer	SSP	USP
Instructions	All	Restricted set
Entered by:	Exception processing	Supervisor program changing mode to user

Activity	Items Saved on Stack
Subroutine call	(PC) [31:0]
Interrupt or trap	(PC) [31:0] (SR) [15:0]
Address error or bus error	(PC) [31:0] (SR) [15:0] Instruction register [15:0] Access address [31:0] Access information [15:0]

Notes:

1. Access address is the address that was being accessed when the error occurred.
2. Access information:
   [2:0]: function code
   [3:3]: 0 = instruction, 1 = not an instruction
   [4:4]: 0 = write, 1 = read
3. The MC68010 has a slightly different stack format, as described in the text.

Supervisor vs User Mode

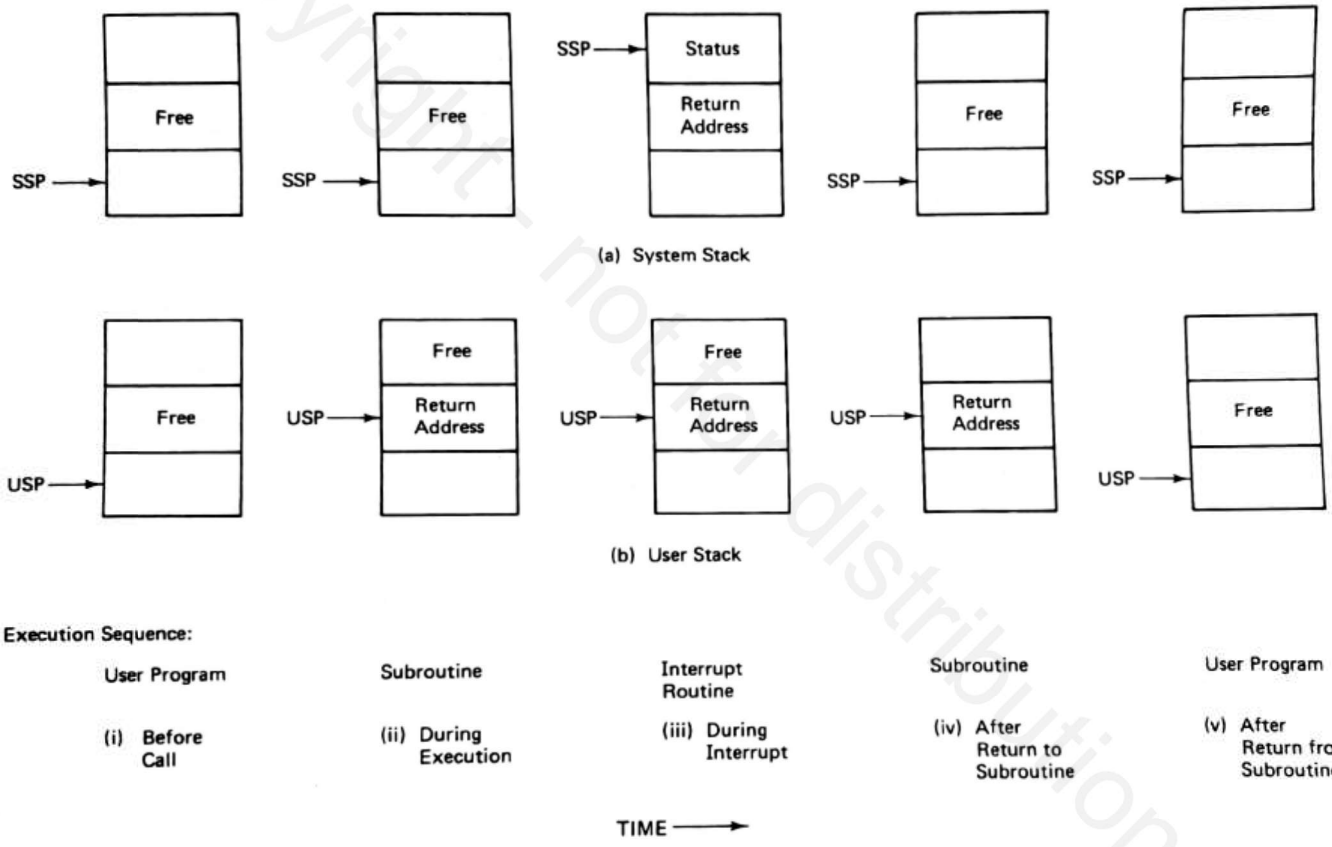


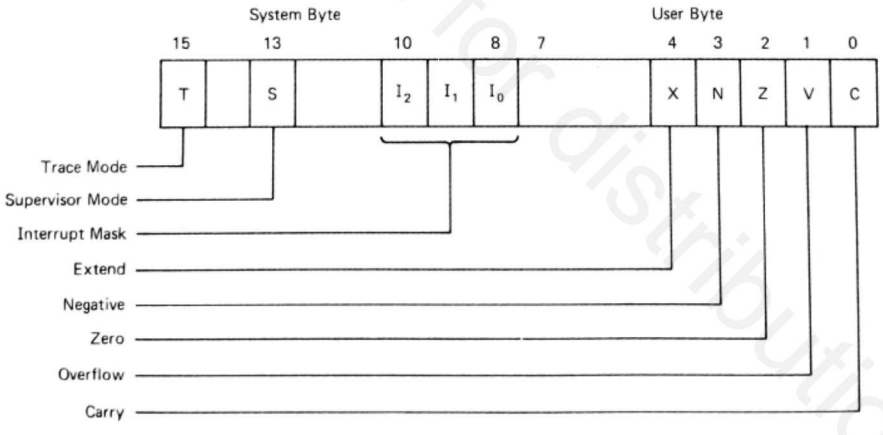
Figure 4.8 Stack usage during execution of a user-made program.

Program Counter

- Program Counter (PC) is a CPU register contains the address of the next instruction to be executed from memory
- Normal sequence of program execution will be every 2 bytes (MC68K 1 word is 16-bit)
  - Next PC = Current PC + 2
- Can be altered by a jump or branch
  - If needs to return, you need to save the next PC in where?

Status Register

- Information about the state of the program
- Mode, interrupt status, arithmetic/logical conditions from last instruction
- 16-bit status register Status Register (SR) only modified in supervisor mode



Notes:
(1) Conditions stated are true when the corresponding bit = {1}.
(2) The user byte portion of the status register is referred to as the condition code register (CCR).

Status Register

Table 4.7 Interpretation of Condition Codes

Name	Symbol	Meaning
Extend	X	Used in multiple-precision arithmetic operations; in many instructions it is set the same as the C bit
Negative	N	Set to {1} if the most significant bit of an operand is {1}
Zero	Z	Set to {1} if all the bits of an operand are {0}
Overflow	V	Set to {1} if an out-of-range condition occurs in two's-complement operations
Carry	C	Set to {1} if a carry is generated out of the most significant bit of the sum in addition; set to {1} if a borrow is generated in subtraction

Table 4.8 Interpretations of System Status

Name	Symbol	Meaning
Trace	T	Set to {1} if the trace mode is being used (single-instruction stepping)
Supervisor	S	Set to {1} if the processor program is in the supervisor mode
Interrupt mask	I0, I1, I2	Coded interrupt level; interrupts at level indicated and below will not be recognized (levels 1–6); level 7 is not maskable