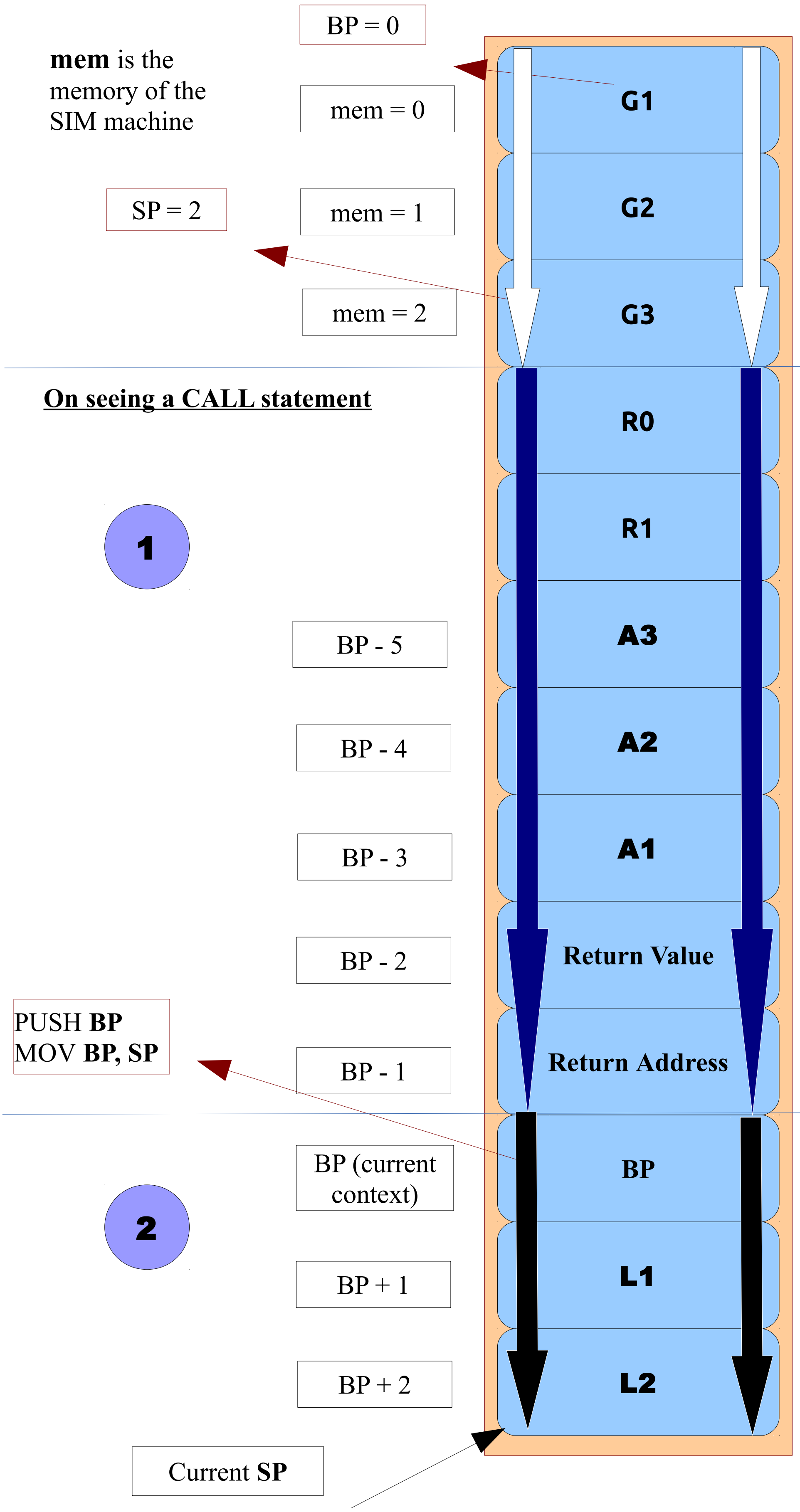
STACK Manipulation on implementing FUNCTIONS in SIL

Mevin M Dominic
B110468CS

Actions to be done before jumping to main

mem is the memory of the SIM machine



This is done initially, ie when the Program starts all the global Variables are pushed on to the stack Starting from memory location 0.

On seeing a CALL statement

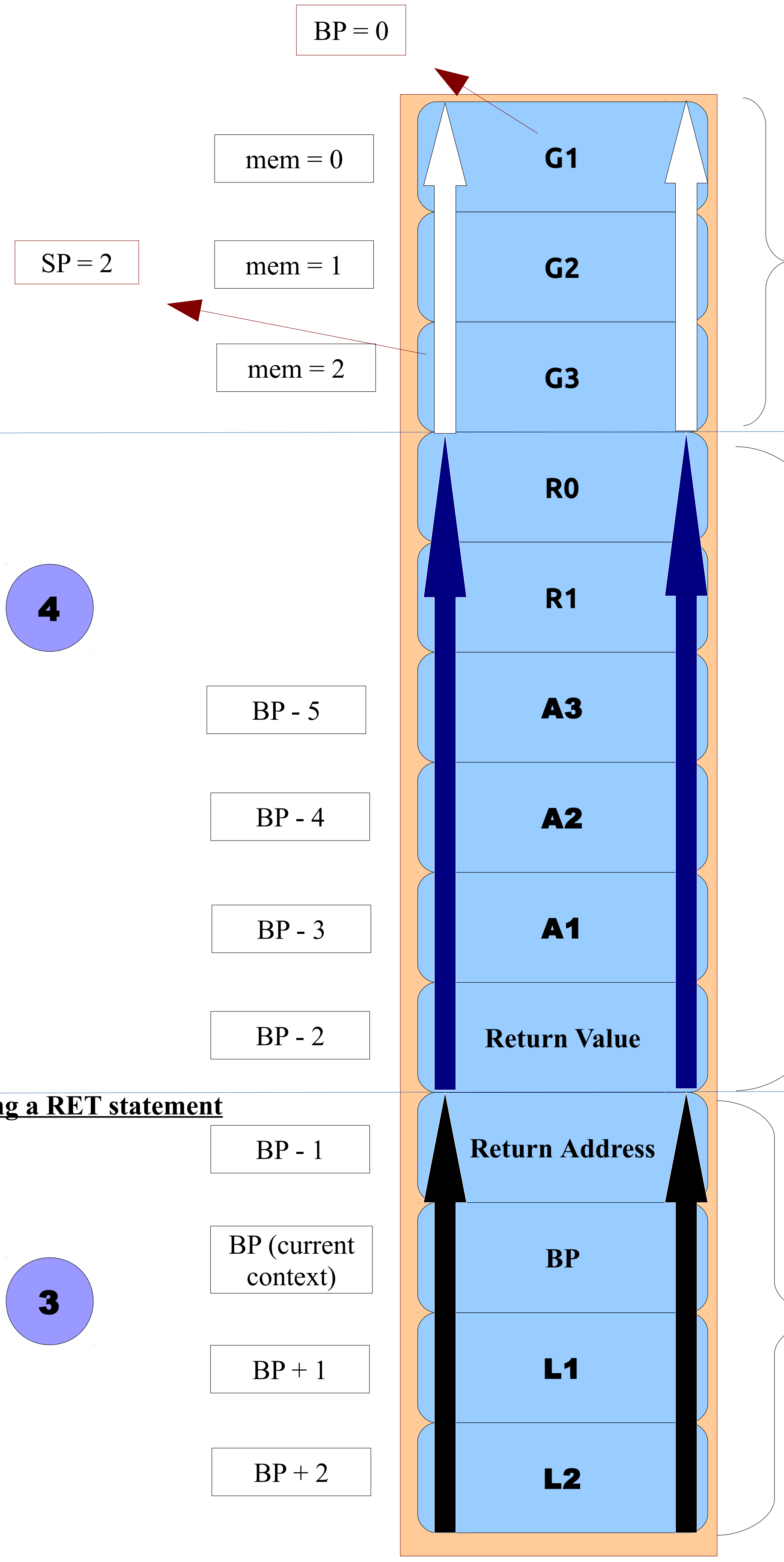
Actions done by the **CALLER** on a call statement,

1) It pushes the registers in use
2) and also the actual arguments of the function call.
3) It reserves space for RETURN Value (manual)and RETURN Address (this is done by the CALL Statement).

EG: Suppose the function call is **func(A1,A2,A3)**. The corresponding stack created is shown on the left.

Actions done by **CALLE** On entry,

1) Saves context (pushes BP) And resets BP to SP.
2) Pushes the local variables L1, L2,..etc.



The initially pushed global variables are cleared off the stack when the Program terminates.

Actions done by a **CALLER** on Return,

1) Obtains the return value
2) Pops off the arguments but only after a check for reference type. If so, the current BP is used to restore the values.
3) Now restore the register values, be careful of the order.
4) While restoring don't lose the Return value which was Obtained in a different register.

On seeing a RET statement

Actions done by **CALLEE** on encountering a return,

1) Sets the **RETURN VALUE** at (**BP-2**).
2) It clears the stack content, ie It pops off local variables.
3) BP of the caller is also obtained.
4) Print the code for the function.
5) The return address is restored on the **RET** command.