

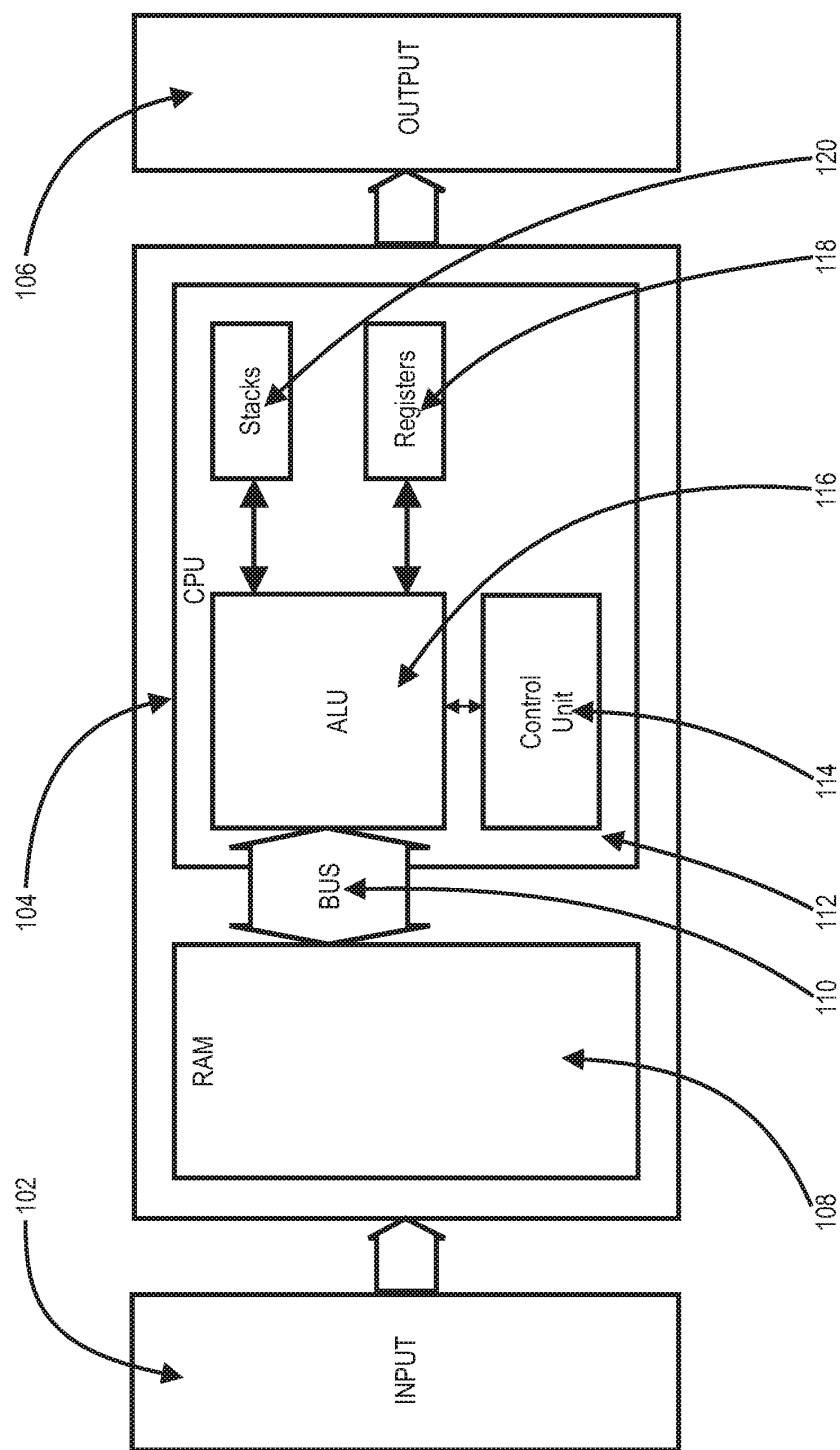


FIG. 1

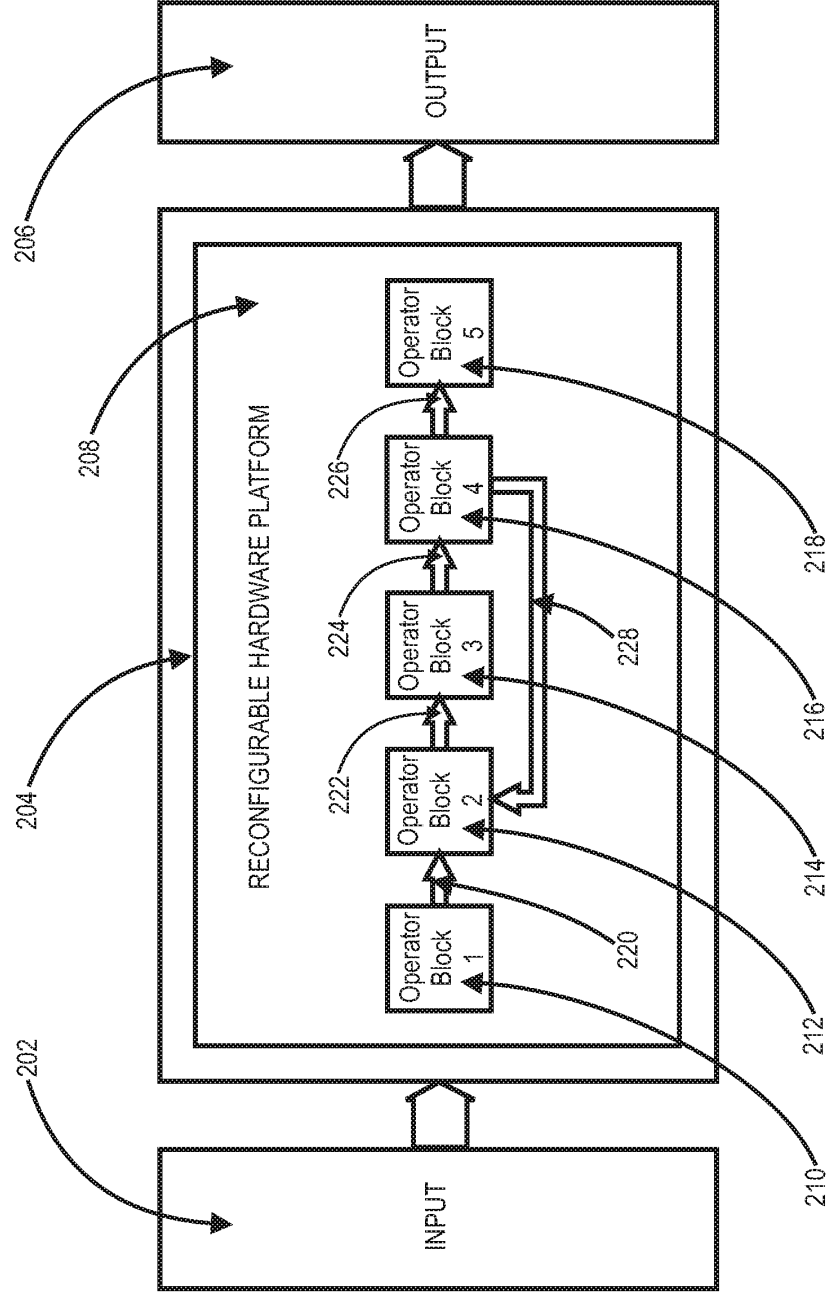


FIG. 2

```
% cat fib.c
#include <stdio.h>

int main (void) {
    int x, y, z;

    While (1) {
        x = 0;
        y = 1;

        do {
            printf("%d\n",
                z = x + y;
                x = y;
                y = z;
            } while (x < 255);
        }
    }
```

FIG. 3

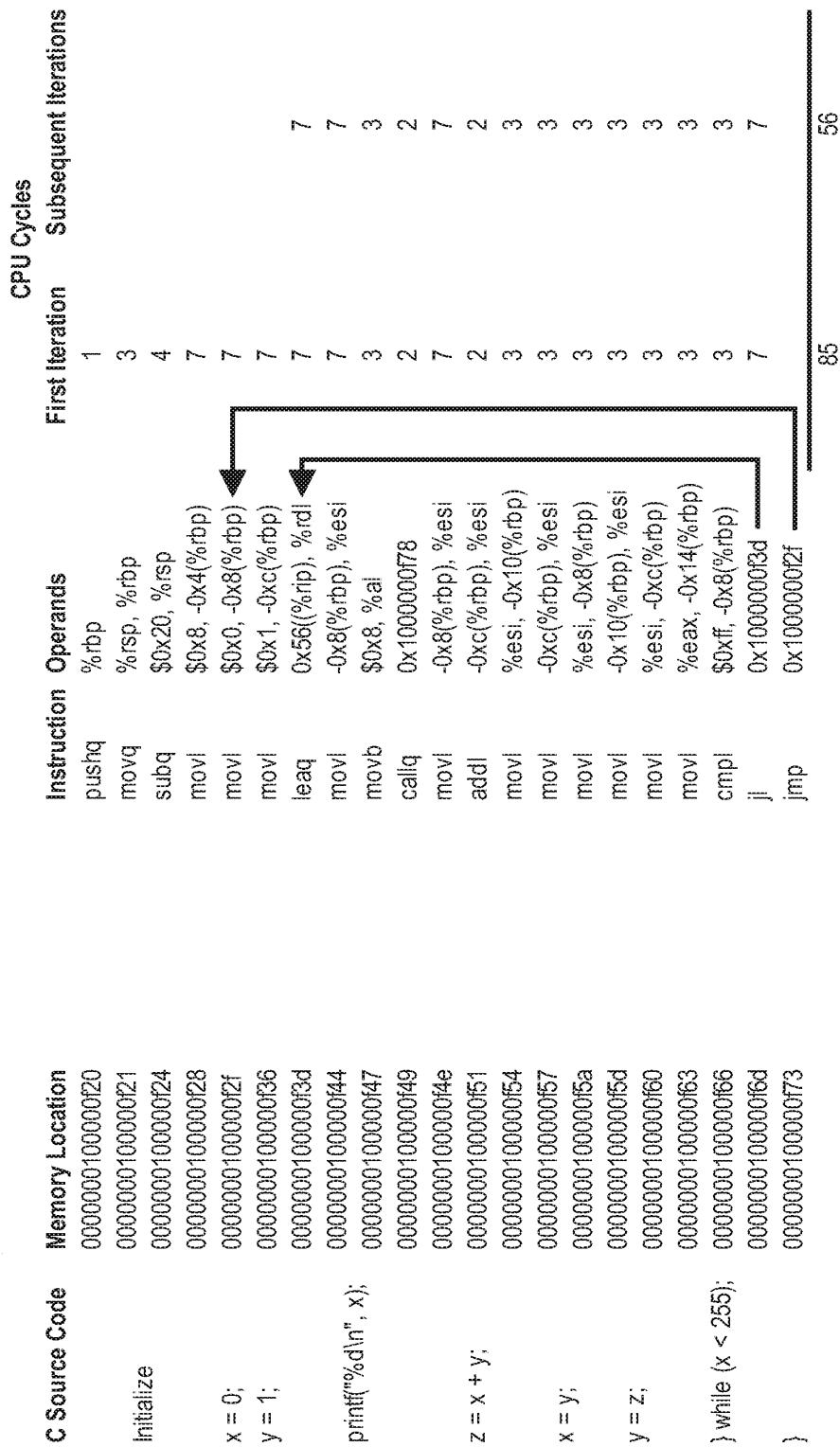


FIG. 4

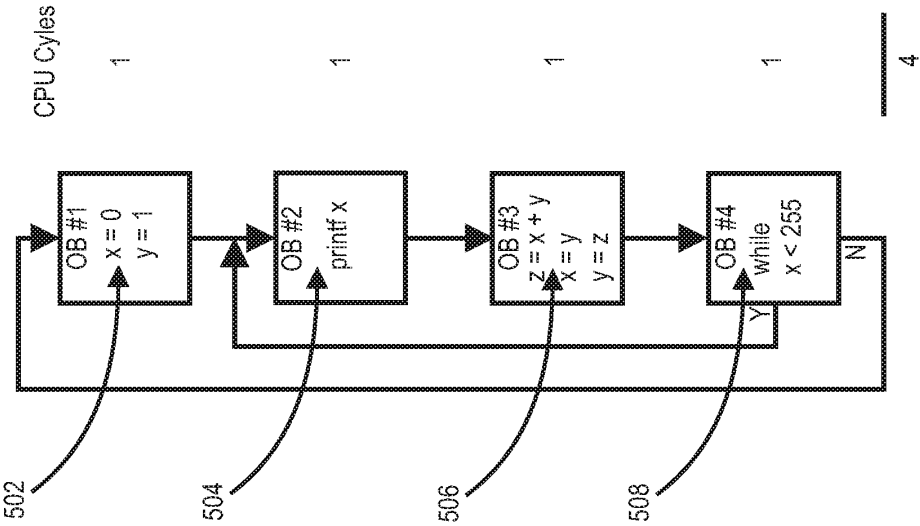


FIG. 5

```
#include <stdio.h>
int sum (int a);

int main()
{
    int num, result;
    printf("Enter the number: ");
    scanf("%d", &num);
    result = sum(num);
    printf("Sum of digits in %d is %d\n", num, result);
    return 0;
}

int sum (int num)
{
    if (num != 0)
    {
        return (num % 10 + sum (num / 10));
    }
    else
    {
        return 0;
    }
}
```

FIG. 6

C Source Code	Memory Location	Instruction	Operands	Cycles	
				If Statement = True	If Statement = False
Initialize		.LC0:			
		.string "Enter the number: "			
		.LC1:			
		.string "%d"			
		.LC2:			
		.string "Sum of digits in %d is %d\n"			
		main:			
	4005d2	push rbp	rbp	1	1
	4005d3	mov	rbp, rsp	3	3
	4005d6	sub	rsp, 16	4	4
	4005da	mov	edi, OFFSET FLAT:.	5	5
	4005df	mov	eax, 0	5	5
	4005e4	call	printf	5	5
	4005e9	lea	rax, [rbp-8]	4	4
	4005ed	mov	rsi, rax	2	2
	4005f0	mov	edi, OFFSET FLAT:.	5	5
	4005f5	mov	eax, 0	5	5
	4005fa	call	scanf	5	5
{					
printf("Enter the number: ");					
scanf("%d", &num);					

FIG. 7

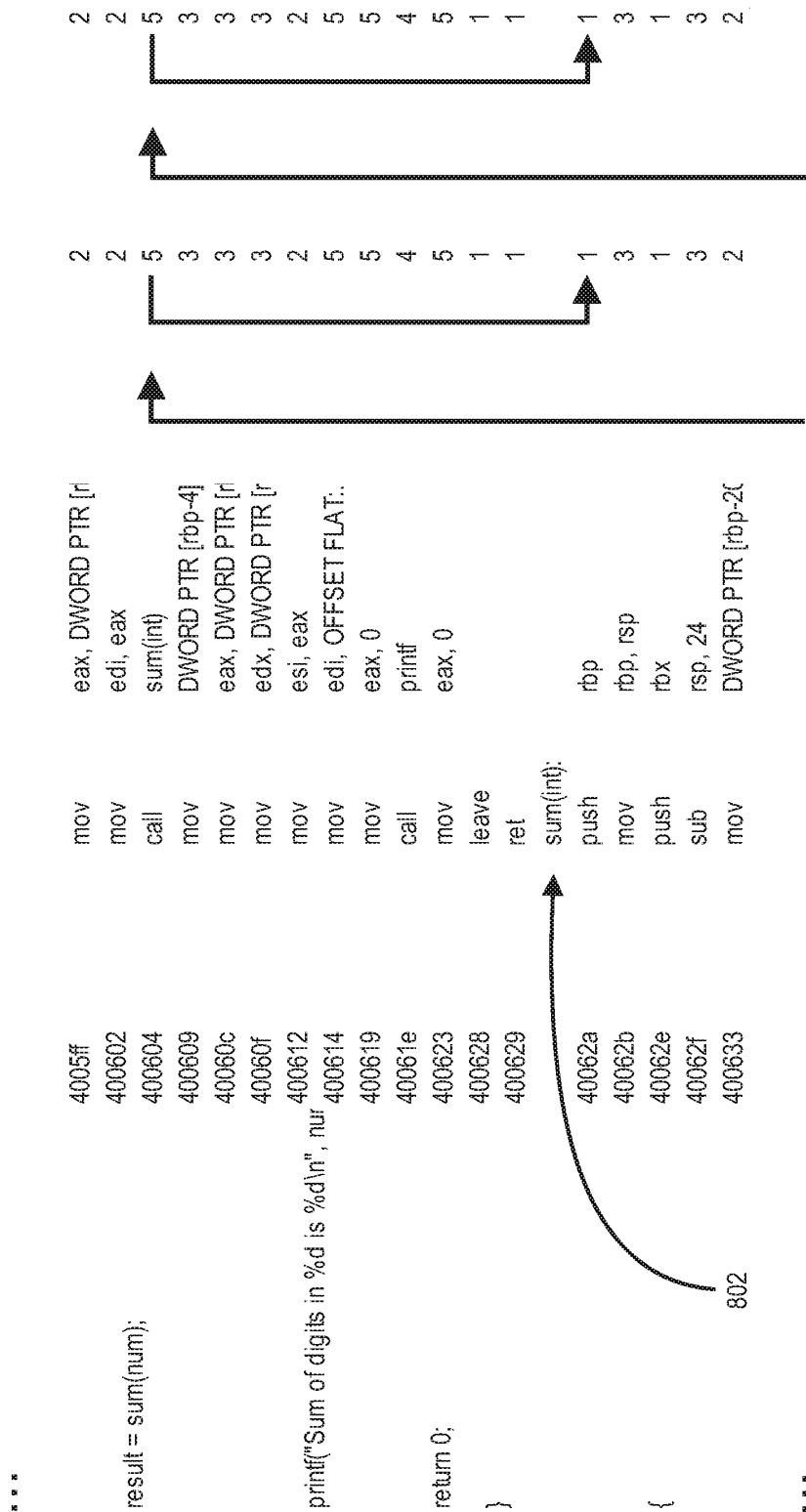


FIG. 8

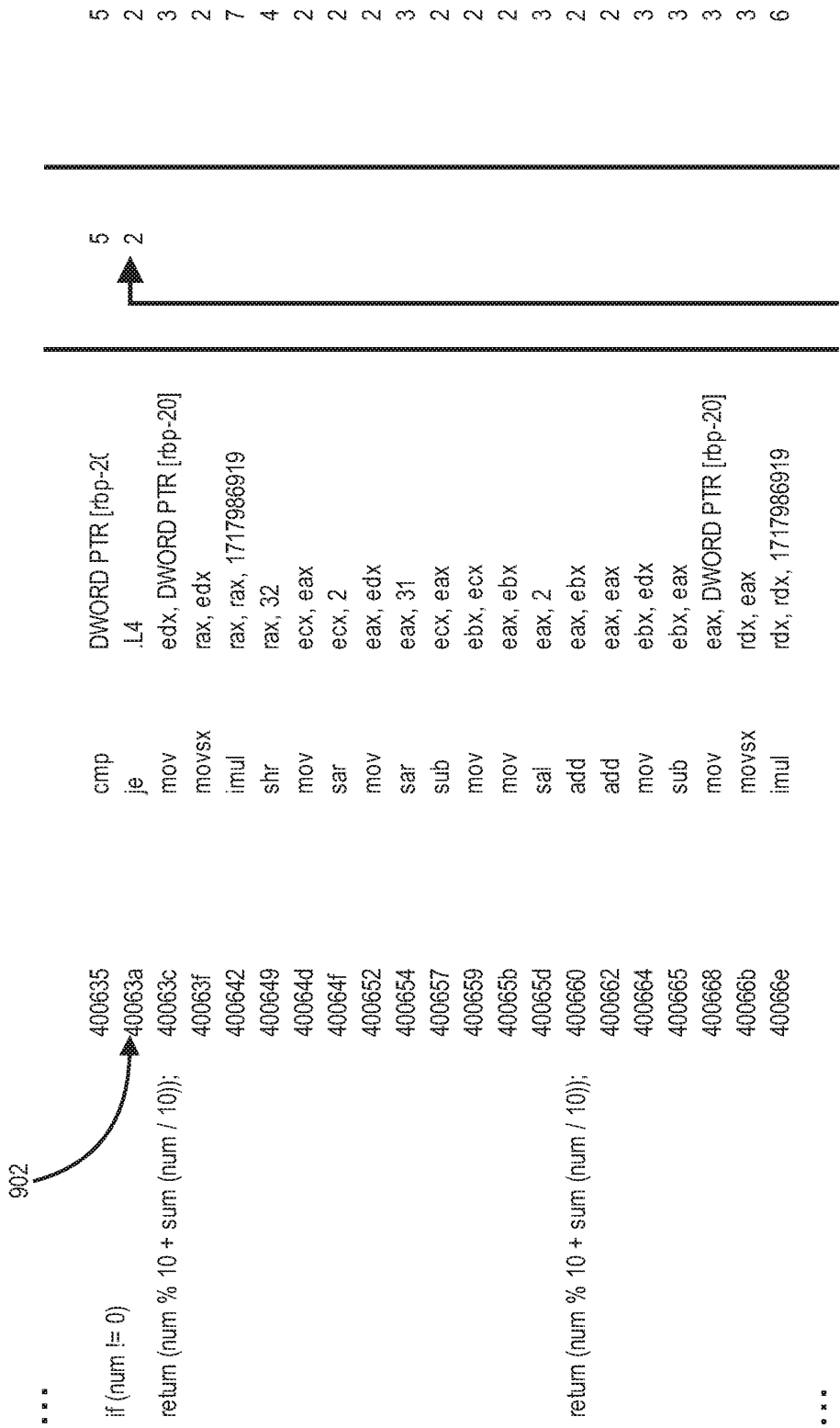


FIG. 9

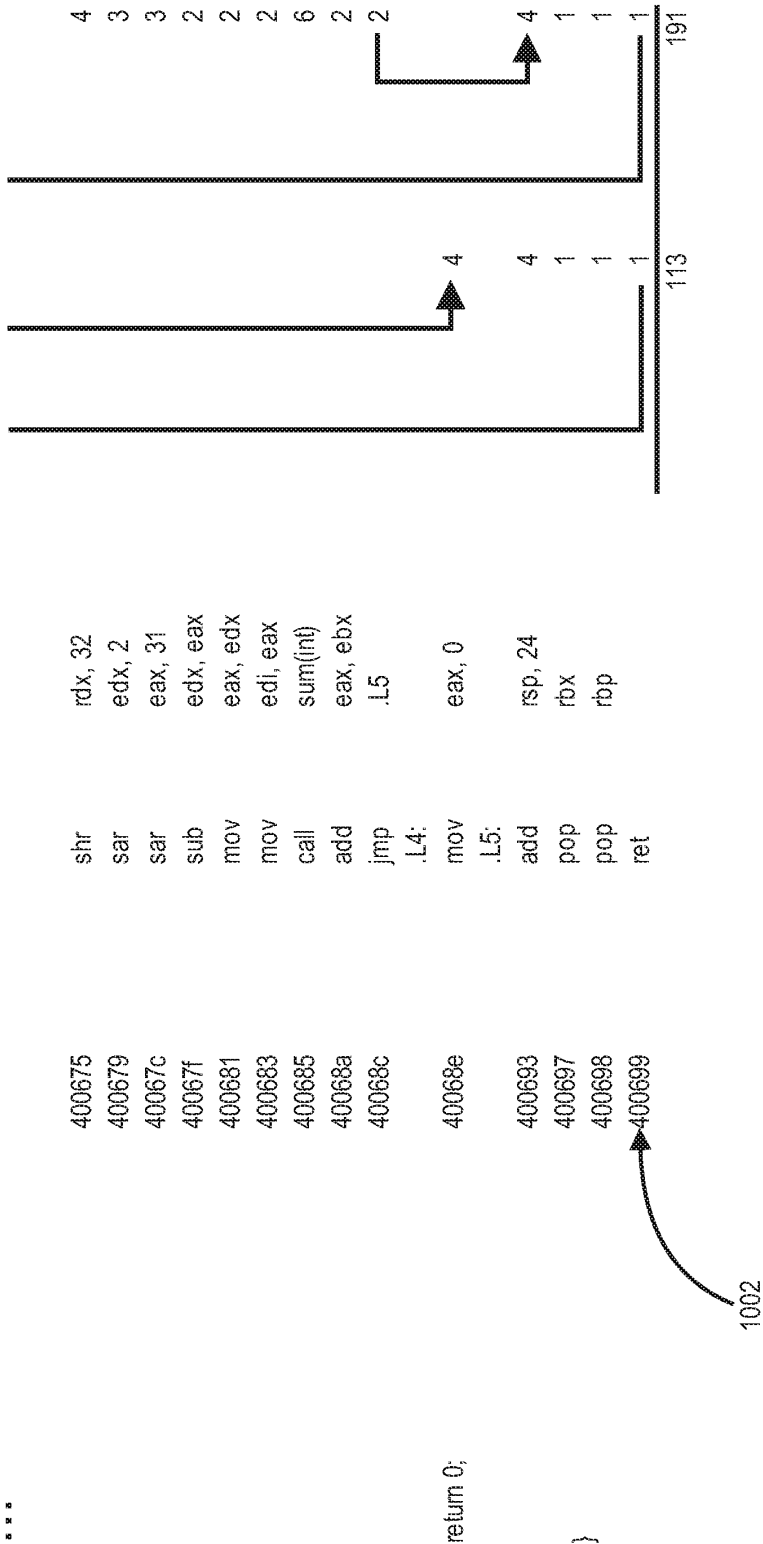


FIG. 10

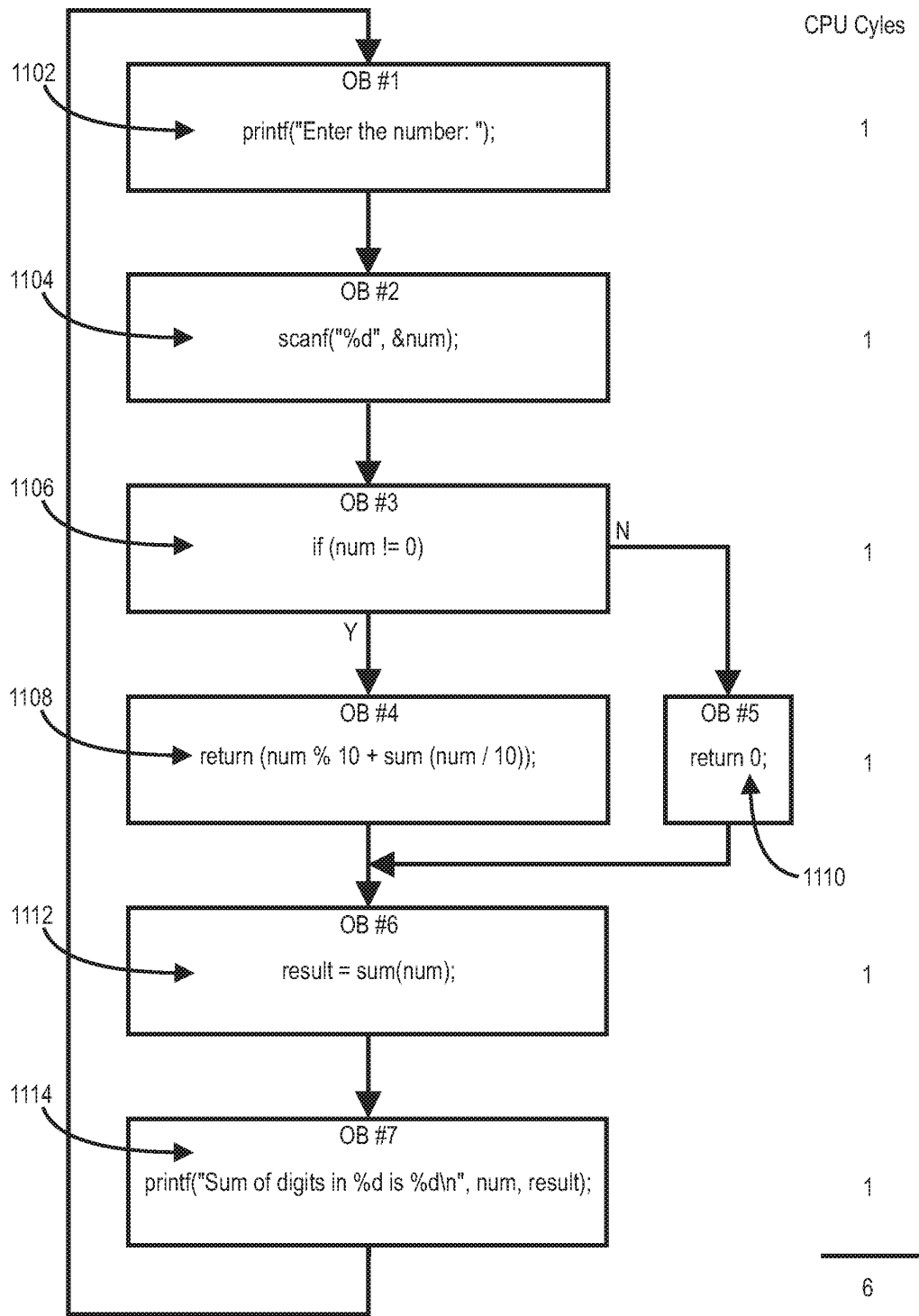
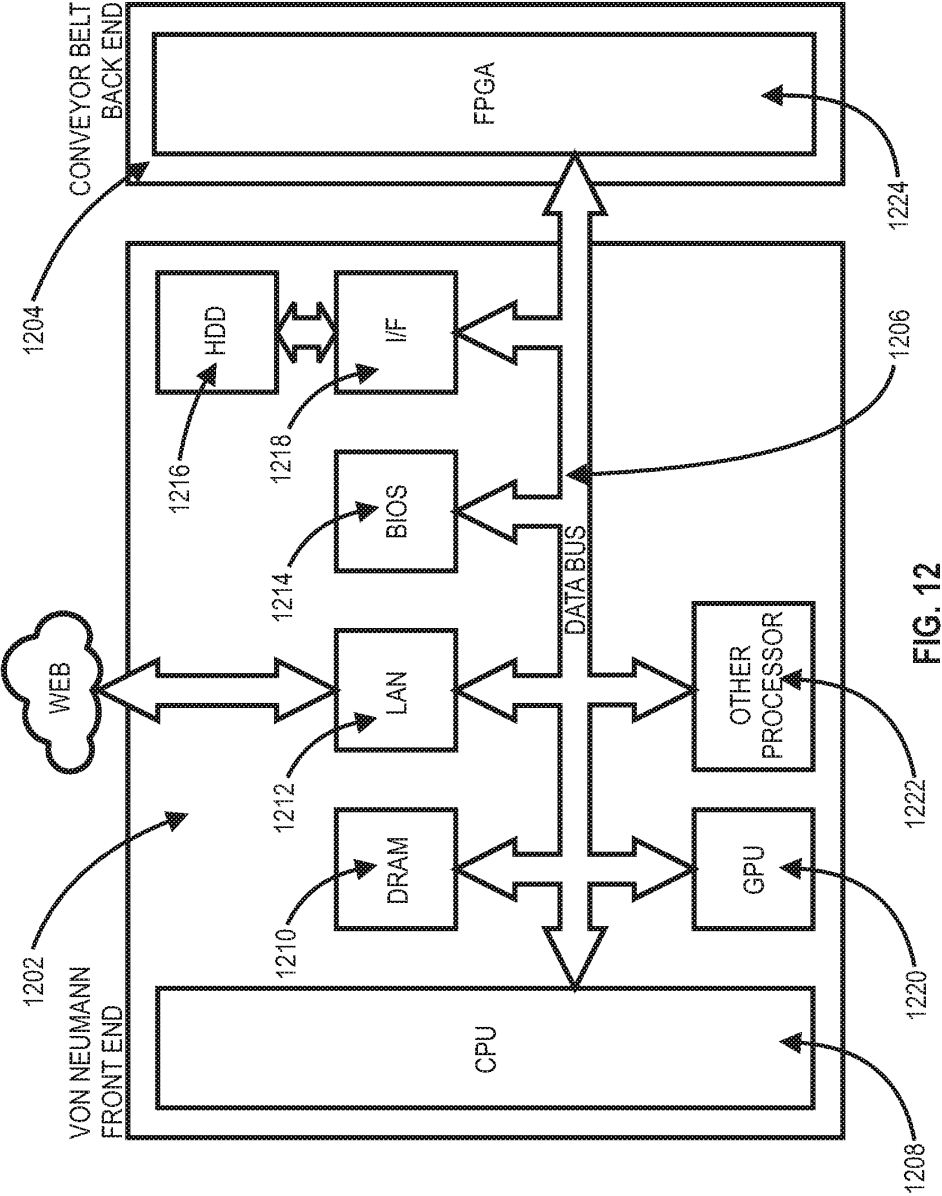


FIG. 11



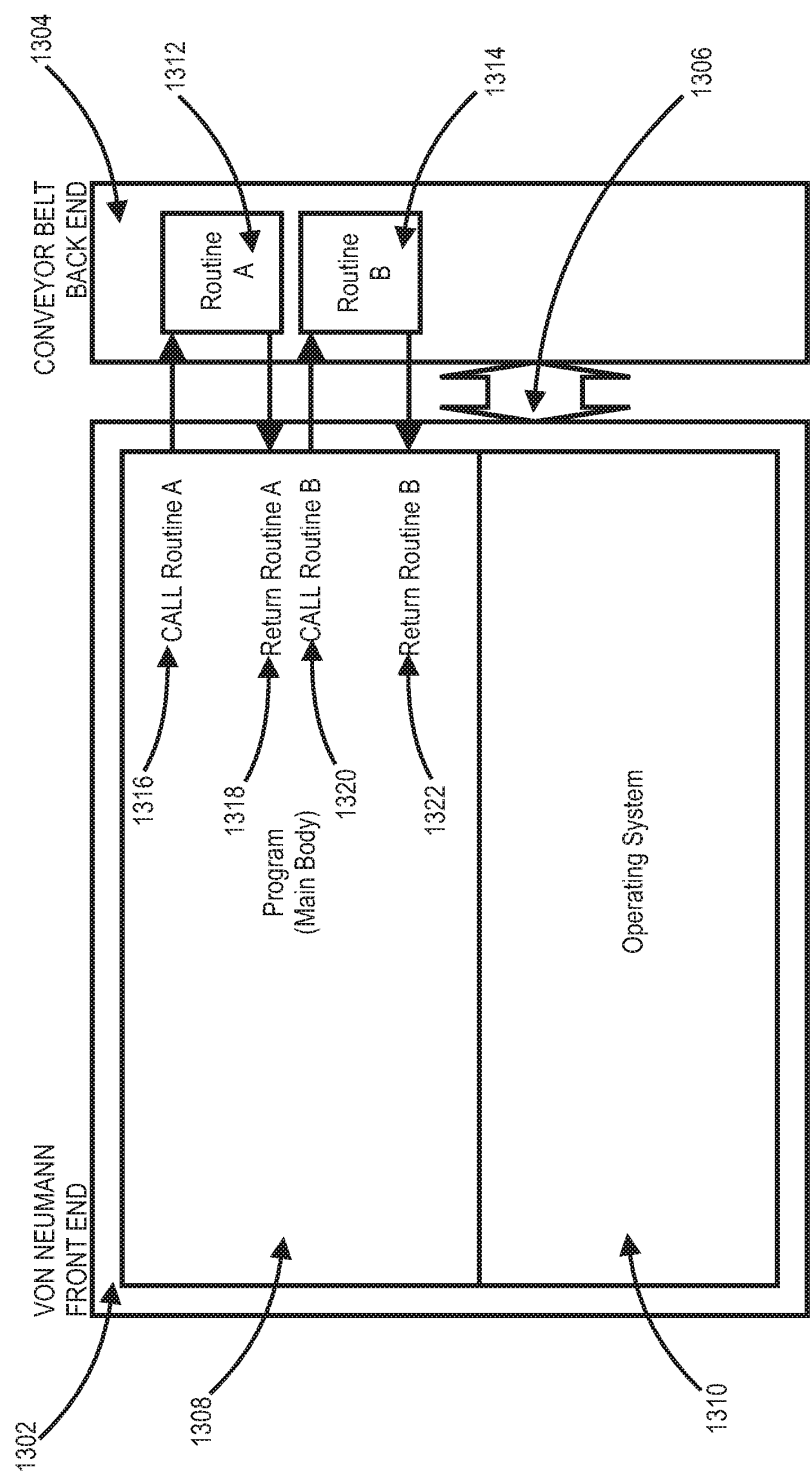
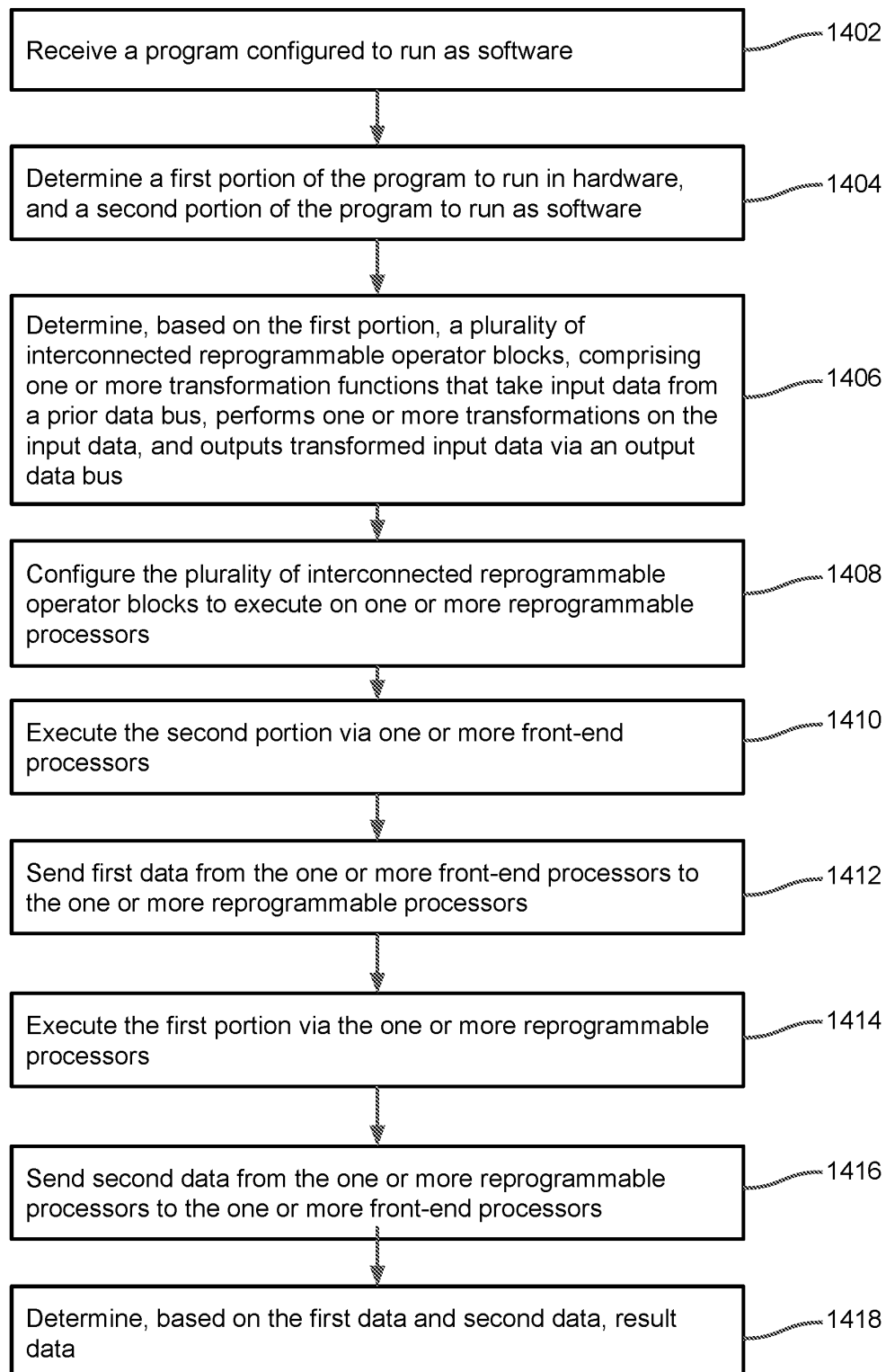


FIG. 13

**FIG. 14**

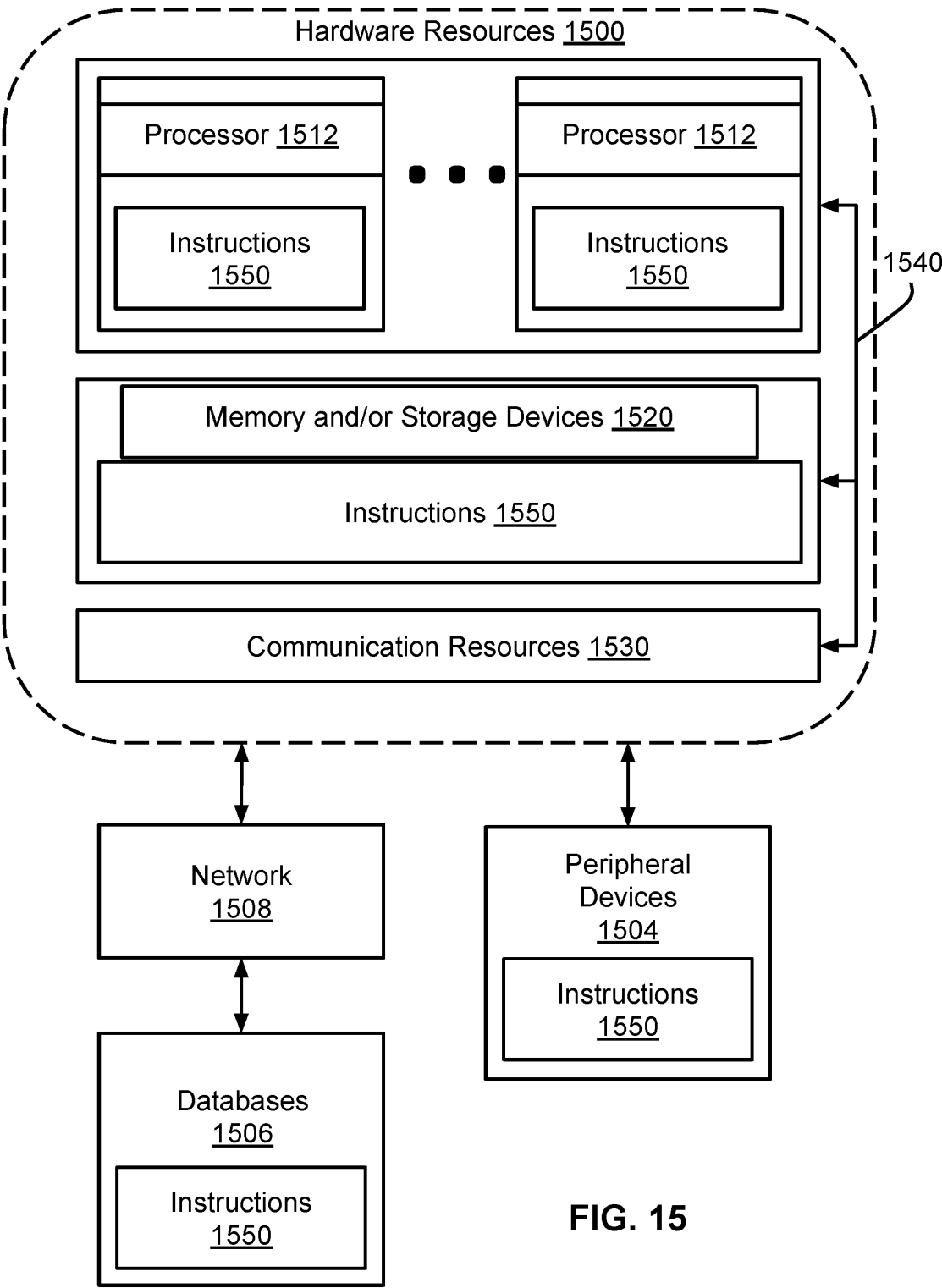


FIG. 15