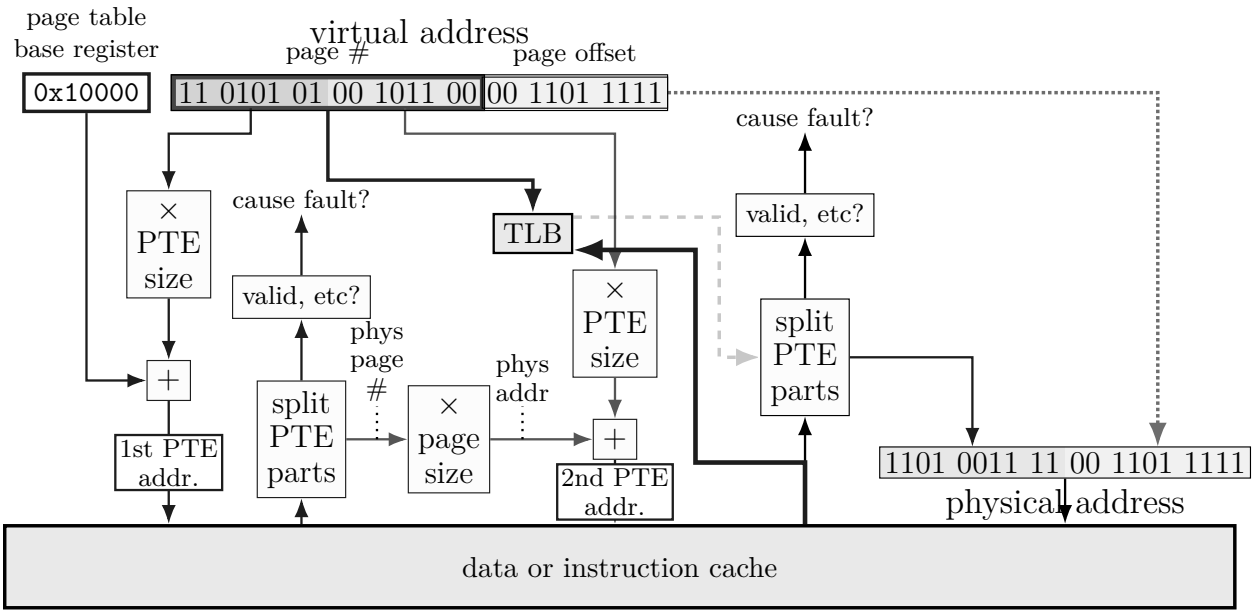
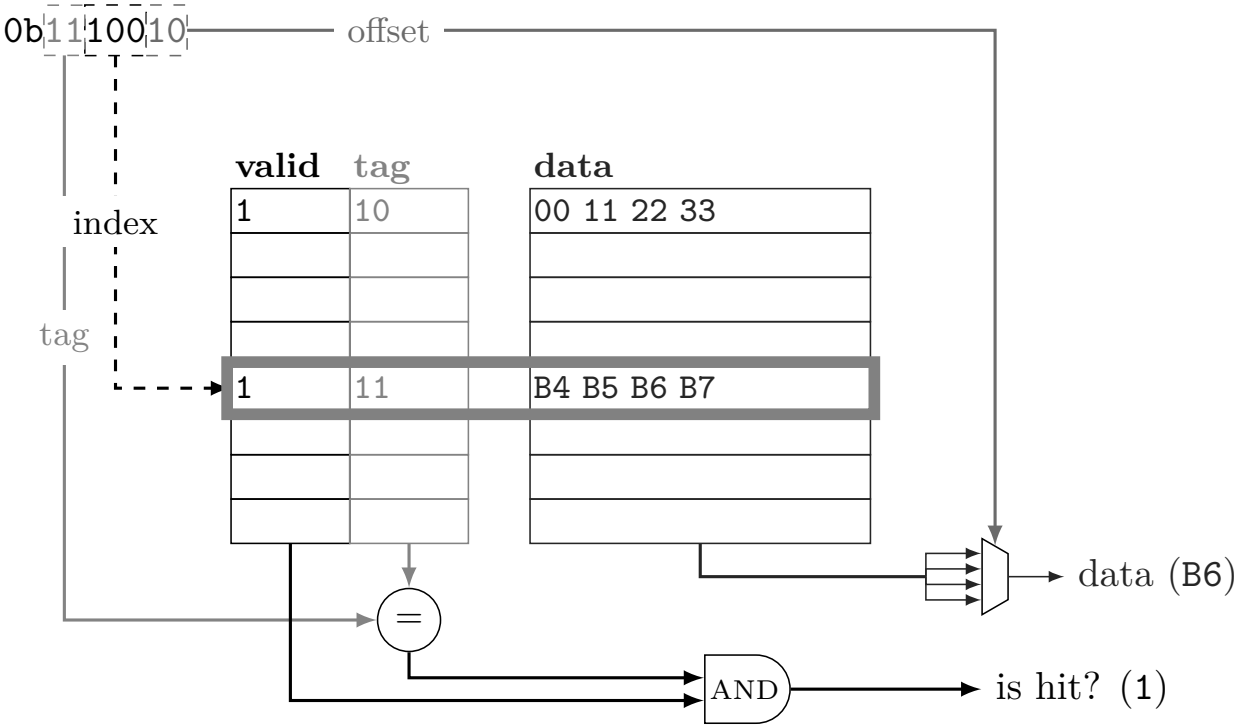
1 page table lookup



2 direct-mapped cache operation



3 selected POSIX functions

- create new process copying current: `fork()` — return new pid in parent (old), 0 in child (new)
- `pipe(fds)` — create a pipe, set fds[0] to the file descriptor for the read end, fds[1] for the write end
- `write(fd, buffer, size)` write `size` bytes from `buffer` to the file descriptor `fd`
- `read(fd, buffer, size)` read up to `size` bytes from `buffer` to the file descriptor `fd`, return total bytes read or 0 on end-of-file
- `dup2(from_fd, to_fd)` makes `to_fd` refer to the same open file as `from_fd`
- `waitpid(pid, 0, NULL)` wait for the child process with ID `pid` to terminate
- `kill(pid, signal_number)` — send signal `signal_number` to process `pid`
- `sigaction(signal_number, &act_struct, NULL)` — configure signal handler for the specified signal based on the information in `act_struct`

4 make syntax

```
target: prerequisites
    command
    command
    ...
```

- default rule is first rule
- pattern rules: use `%` to represent base of filename (e.g. `%.o: %.c` for x.o from x.c)
- selected automatic variables:

`$@`	target of current rule
`$<`	first prerequisite of current rule
`$^`	all prerequisites of current rule
`$*`	'stem' of a pattern rule (what the % matched)