Латентность при обращении к памяти.

Латентностью используется при описании конвейерных структур. Латентностью можно назвать время, за которое некая сущность пройдёт полный путь на конвейере и будет полностью “собрана”. Латентность при обращении к памяти измеряется в тактах процессора. Логично, что, чем больше размер адресуемой памяти, тем меньше скорость обращения. Латентность при обращении к регистрам процессора – 1. Память на это – примерно 1 кбит.

Для регистров не нужно большое их количество, они преимущественно используются для работы с данными.

Далее по скорости работы идут кэши первого, второго и третьего уровней. Кэш первого уровня находится ближе всего к ядру системы и работает с непосредственно необходимыми для вычисления данными. Латентность кэша первого уровня варьируется от 2 до 4 тактов в зависимости от попадания. Размер, приблизительно, 64 кбайта.

Для кэша второго уровня память измеряется в мегабайтах, время доступа уже от 10 до 12 тактов. Стоит отметить, что время доступа для более крупных по размеру структур, мы также вынуждены обращаться к более низкоуровневым, что также сказывается на скорости доступа.

Для третьего кэша латентность составляет от 15 до 50 тактов, память становится большей по размеру.

Если мы хотим обратиться к памяти, расположенной на одном кристалле с системой, то мы обращаемся через внутренние шины и это быстрее, но, всё же, обращение занимает до 75 тактов. Стоит учитывать, что это значение верно при работе через TLB – буффера, позволяющего быстро преобразовать виртуальный адрес в физический.

В случае внешней памяти работа с памятью занимает от 200 до 400 тактов процессора. Это значение актуально для работы к, например, любой переменной в памяти. Это обращение гарантирует больший простой с учётом более низких уровней. В случае промаха в TLB время обработки может быть гораздо больше. (от 2000 до 2500 процессорных тактов при отсутствии адреса физической памяти при промахе).

Для внешних накопителей время может достигать миллионов (возможно, даже миллиардов) тактов.