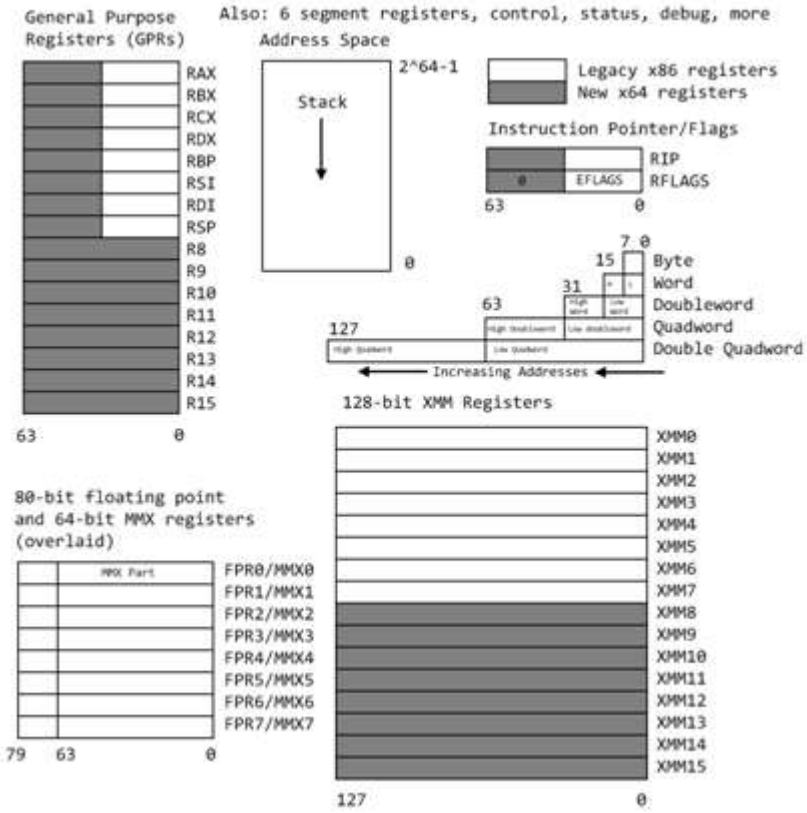
Различия между 32bit и 64bit ассемблером.

Регистры 64 Bit.

Наследованные регистры были расширены до 64bit, при этом доступ до младших бит по отдельному имени остался. Например, EAX это младшие 32 бита регистра RAX. Всего теперь 16 регистров общего назначения. Ниже приведем их на схеме.



Исторически наследованные регистры RAX, RBX,RCX,RDX,RBP,RSI,RDI,RSP расширены дополнительным набором из 8 регистров от R8 до R15. Это регистры новые и не имеют аналогов в 32bit архитектуре. Регистры математического сопроцессора остались без изменений, а вот регистры XMM дополнены ещё 8 регистрами, от XMM8 до XMM15.

Есть странное ограничение – нельзя использовать в одной команде старшие байтовые регистры (AH,BH,CH,DH) и новые байтовые регистры (R8B – R15B), однако их можно использовать вместе с младшими байтовыми регистрами (AL,BL,CL,DL). Это вынужденная мера из-за замены регистров AH,BH,CH,DH регистрами BPL,SPL,DIL,SIL в инструкциях, использующих префикс REX.

Регистр ESP стал 64-разрядным и именуется RSP. Функционал его не изменился.

Регистр EIP стал 64-разрядным и именуется RIP. Функционал его не изменился.

Регистр RFLAGS дополнен старшими 32 битами, которые сейчас не используются и зарезервированы.

Команды 64bit.

Команды полностью унаследованы, исключенных команд нет. Команды дополнены 64bit вариантом, например IDIV RBX выполняет целочисленное деление 128bit значения из пары регистров RDX:RAX на RBX . Результат помещается в RAX, остаток от деления в RDX.

Память 64bit.

На текущий момент (2021 год) все 2^{64} адресное пространство ни одним процессором и ОС не поддерживалось. Процессоры понимали 2^{48}, а например ОС Windows не могла адресовать более чем 2^{44} (16 терабайт) памяти. Ограничения связаны с моделью памяти и виртуального адресного пространства Windows.

Соглашения и использование стека.

Операции со стеком используют 64bit разрядность вне зависимости от отправляемого в стек значения. При помещении в стек значения меньше 64bit старшие байты не заполняются нулями и содержат мусор. При возвращении из стека рекомендуется следующее : целые числа возвращать в RAX, числа с плавающей точкой в XMM0.

При программировании в регистрах RCX,RDX,R8 и R9 располагать целые числа и указатели, а в XMM0, XMM1,XMM2 и XMM3 числа с плавающей точкой.

При работе с процедурами регистры RAX, RCX, RDX, R8, R9, R10, R11 считать утратившими значение, а значения регистров RBX, RBP, RDI, RSI, R12, R13, R14, R15 сохранять при входе в функцию и восстанавливать при выходе.

Пример. Hello world.

```
include "win64ax.inc"

.data

Caption db 'Win64 assembly program',0

Message db 'Hello World!',0

.code
```

```
start:
xor r9d,r9d
lea r8,[Caption]
lea rdx,[Message]
xor rcx,rcx
call [MessageBox]
mov ecx,eax
invoke ExitProcess,0
.end start
```