`asm_experimental_arch`

The tracking issue for this feature is: #93335

This feature tracks `asm!` and `global_asm!` support for the following architectures:

- NVPTX
- PowerPC
- Hexagon
- MIPS32r2 and MIPS64r2
- wasm32
- BPF
- SPIR-V
- AVR
- MSP430

Register classes

Architecture	Register class	Registers	LLVM constraint code
MIPS	`reg`	`$[2-25]`	`r`
MIPS	`freg`	`$f[0-31]`	`f`
NVPTX	`reg16`	None*	`h`
NVPTX	`reg32`	None*	`r`
NVPTX	`reg64`	None*	`l`
Hexagon	`reg`	`r[0-28]`	`r`
PowerPC	`reg`	`r[0-31]`	`r`
PowerPC	`reg_nonzero`	`r[1-31]`	`b`
PowerPC	`freg`	`f[0-31]`	`f`
PowerPC	`cr`	`cr[0-7]`, `cr`	Only clobbers
PowerPC	`xer`	`xer`	Only clobbers
wasm32	`local`	None*	`r`
BPF	`reg`	`r[0-10]`	`r`
BPF	`wreg`	`w[0-10]`	`w`
AVR	`reg`	`r[2-25]`, `XH`, `XL`, `ZH`, `ZL`	`r`
AVR	`reg_upper`	`r[16-25]`, `XH`, `XL`, `ZH`, `ZL`	`d`
AVR	`reg_pair`	`r3r2` .. `r25r24`, `X`, `Z`	`r`
AVR	`reg_iw`	`r25r24`, `X`, `Z`	`w`
AVR	`reg_ptr`	`X`, `Z`	`e`

MSP430	`reg`	`r[0-15]`	`r`

> ***Notes***:
>
> - *NVPTX doesn't have a fixed register set, so named registers are not supported.*
> - *WebAssembly doesn't have registers, so named registers are not supported.*

Register class supported types

Architecture	Register class	Target feature	Allowed types
MIPS32	`reg`	None	`i8`, `i16`, `i32`, `f32`
MIPS32	`freg`	None	`f32`, `f64`
MIPS64	`reg`	None	`i8`, `i16`, `i32`, `i64`, `f32`, `f64`
MIPS64	`freg`	None	`f32`, `f64`
NVPTX	`reg16`	None	`i8`, `i16`
NVPTX	`reg32`	None	`i8`, `i16`, `i32`, `f32`
NVPTX	`reg64`	None	`i8`, `i16`, `i32`, `f32`, `i64`, `f64`
Hexagon	`reg`	None	`i8`, `i16`, `i32`, `f32`
PowerPC	`reg`	None	`i8`, `i16`, `i32`
PowerPC	`reg_nonzero`	None	`i8`, `i16`, `i32`
PowerPC	`freg`	None	`f32`, `f64`
PowerPC	`cr`	N/A	Only clobbers
PowerPC	`xer`	N/A	Only clobbers
wasm32	`local`	None	`i8` `i16` `i32` `i64` `f32` `f64`
BPF	`reg`	None	`i8` `i16` `i32` `i64`
BPF	`wreg`	`alu32`	`i8` `i16` `i32`
AVR	`reg`, `reg_upper`	None	`i8`
AVR	`reg_pair`, `reg_iw`, `reg_ptr`	None	`i16`
MSP430	`reg`	None	`i8`, `i16`

Register aliases

Architecture	Base register	Aliases
Hexagon	`r29`	`sp`

Hexagon	`r30`	`fr`
Hexagon	`r31`	`lr`
BPF	`r[0-10]`	`w[0-10]`
AVR	`XH`	`r27`
AVR	`XL`	`r26`
AVR	`ZH`	`r31`
AVR	`ZL`	`r30`
MSP430	`r0`	`pc`
MSP430	`r1`	`sp`
MSP430	`r2`	`sr`
MSP430	`r3`	`cg`
MSP430	`r4`	`fp`

> ***Notes**:*
>
> - *TI does not mandate a frame pointer for MSP430, but toolchains are allowed to use one; LLVM uses `r4`.*

Unsupported registers

Architecture	Unsupported register	Reason
All	`sp`	The stack pointer must be restored to its original value at the end of an asm code block.
All	`fr` (Hexagon), `$fp` (MIPS), `Y` (AVR), `r4` (MSP430)	The frame pointer cannot be used as an input or output.
All	`r19` (Hexagon)	This is used internally by LLVM as a "base pointer" for functions with complex stack frames.
MIPS	`$0` or `$zero`	This is a constant zero register which can't be modified.
MIPS	`$1` or `$at`	Reserved for assembler.
MIPS	`$26`/`$k0`, `$27`/`$k1`	OS-reserved registers.
MIPS	`$28`/`$gp`	Global pointer cannot be used as inputs or outputs.
MIPS	`$ra`	Return address cannot be used as inputs or outputs.
Hexagon	`lr`	This is the link register which cannot be used as an input or output.
AVR	`r0`, `r1`, `r1r0`	Due to an issue in LLVM, the `r0` and `r1` registers cannot be used as inputs or outputs. If modified, they must be restored to their original values before the end of the block.

MSP430	`r0`, `r2`, `r3`	These are the program counter, status register, and constant generator respectively. Neither the status register nor constant generator can be written to.

Template modifiers

Architecture	Register class	Modifier	Example output	LLVM modifier
MIPS	`reg`	None	`$2`	None
MIPS	`freg`	None	`$f0`	None
NVPTX	`reg16`	None	`rs0`	None
NVPTX	`reg32`	None	`r0`	None
NVPTX	`reg64`	None	`rd0`	None
Hexagon	`reg`	None	`r0`	None
PowerPC	`reg`	None	`0`	None
PowerPC	`reg_nonzero`	None	`3`	`b`
PowerPC	`freg`	None	`0`	None

Flags covered by `preserves_flags`

These flags registers must be restored upon exiting the asm block if the `preserves_flags` option is set:

- AVR
    - The status register `SREG`.
- MSP430
    - The status register `r2`.